**Fast Differential-Geometric Methods for Continuous Muscle Wrapping Over Multiple General Surfaces**

Von der Fakultät für Ingenieurwissenschaften, Abteilung Maschinenbau und Verfahrenstechnik der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades eines

Doktoringenieurs
Dr.-Ing.

genehmigte Dissertation
von

Andreas Scholz
aus
Duisburg

| | | |
|---|---|---|
| Referent | : | Prof. Dr. Andrés Kecskeméthy |
| Korreferent | : | Prof. Mag. Dr. Dr. h. c. Manfred Husty |

Tag der mündlichen Prüfung: 30. November 2015

# Vorwort - Preface

Die vorliegende Dissertationsschrift ist während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Mechanik und Robotik der Universität Duisburg-Essen entstanden.

Der Ursprung der hier behandelten Fragestellung nach der lokal kürzesten Kurve zwischen zwei Punkten entlang mehrerer geometrischer Hindernisflächen liegt jedoch noch vor dieser Zeit, nämlich zu Beginn meines Forschungsaufenthaltes am Neuromuscular Biomechanics Lab in Stanford zur Anfertigung meiner Masterarbeit. Schnell zeigte sich, dass eine erschöpfende Bearbeitung des Themas den Rahmen einer Masterarbeit bei weitem übersteigen würde.

Ich möchte an dieser Stelle die Gelegenheit nicht versäumen jenen Menschen zu danken, die diese Arbeit ermöglicht und unterstützt haben. Mein besonderer Dank gebührt Prof. Andrés Kecskeméthy für die wissenschaftlich ausgezeichnete Betreuung und die einzigartige Liebe zur Thematik.

Mein herzlicher Dank gilt auch Prof. Scott Delp und Michael Sherman vom Neuromuscular Biomechanics Lab für die Formulierung dieser interessanten Problemstellung sowie für die angenehme Zusammenarbeit sowohl während meines Forschungsaufenthaltes als auch danach.

Bedanken möchte ich mich ebenfalls herzlich bei Prof. Manfred Husty für sein sorgfältiges Gutachten zu dieser Dissertation sowie die wertvollen Anregungen.

Während meiner gesamten Zeit am Lehrstuhl hatte ich die Freude mit vielen tollen Kollegen zusammenzuarbeiten. Es waren die kleinen Gespräche in der Mittagspause oder im Büro, aber auch die gemeinsamen Treffen nach Feierabend, die eine besondere Arbeitsatmosphäre möglich gemacht haben.

Zu guter Letzt gebührt mein innigster Dank meiner Familie und ganz besonders meiner Lebensgefährtin Maike für den Rückhalt und die Unterstützung sowohl während des Studiums als auch während der Promotionszeit.

Andreas Scholz, 2015

Indes sie forschten, röntgten, filmten, funkten,
entstand von selbst die köstlichste Erfindung:
der Umweg als die kürzeste Verbindung zwischen zwei Punkten.

Erich Kästner

# 1 Abstract

Musculoskeletal simulation has become an essential tool for understanding human locomotion and movement disorders. Muscle-actuated simulations require methods that continuously compute musculotendon paths, their lengths, and their rates of length change to determine muscle forces, moment arms, and the resulting body and joint loads. Musculotendon paths are often modeled as locally length-minimizing curves that wrap frictionlessly over moving obstacle surfaces representing bone and tissue. Biologically accurate wrapping surfaces are complex and a single muscle path may wrap around multiple obstacles. However, state-of-the-art muscle wrapping methods are either limited to analytical results for a pair of simple surfaces, or they are computationally expensive.

This thesis describes a new method for the fast and accurate computation of a massless musculotendon's shortest path that wraps frictionlessly across an arbitrary number of general smooth wrapping surfaces. Furthermore, an explicit formula for the path's exact rate of length change is presented, as well as an algorithm for simulating path lift-off and touchdown. The total path is regarded as a concatenation of straight-line segments between local surface geodesics, where each geodesic is naturally parameterized by its start point, direction, and length. The shortest path is computed by finding the root of a vector-valued global path-error constraint equation that enforces that the geodesics connect collinearly with adjacent straight-line segments. High computational efficiency is achieved using Newton's method to zero the path error with an explicit, banded Jacobian that maps natural variations of the geodesic parameters to path-error variations.

Simulation benchmarks demonstrate that the proposed method computes high-precision solutions for path length and rate of length change, and that it allows for wrapping over biologically accurate surfaces that can be described either parametrically or implicitly. By using the explicit path-error Jacobian, the proposed method is very efficient and thus allows for simulating muscle paths over hundreds of surfaces in real time.

# Contents

# 2 Introduction

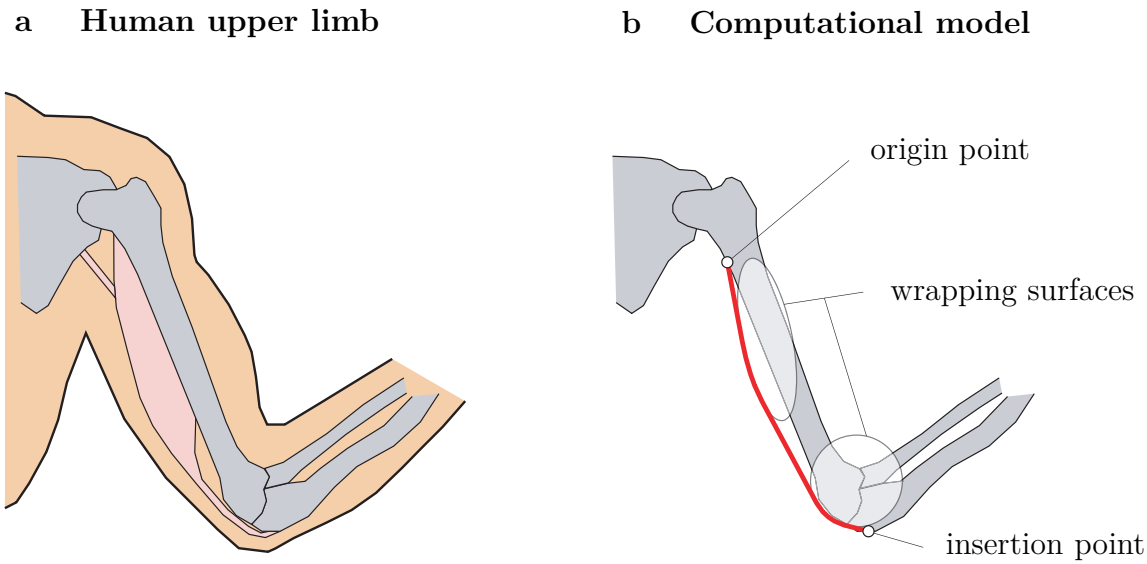
Musculoskeletal simulation has become an essential tool for understanding human locomotion and movement disorders because it provides insights into internal principles of operation of human and animal bodies, which are not ascertainable otherwise. Musculoskeletal models are often developed with a view to investigating how individual muscles contribute to motion, both in the normal and the pathologic case.

The question how individual muscles contribute to motion is inevitably linked with the question which paths muscles and tendons, referred to as muscles for short, take inside the body. This is because of two reasons. First, a muscle's ability to generate force depends on its current length and its current rate of length change. Second, a muscle's path affects both its moment arms relative to the actuated joints as well as the distribution of forces onto the skeleton.

Musculoskeletal models require methods that compute muscle paths, their lengths and their rates of length change to predict the interaction between muscle force and motion. It is a common technique to model muscles as thin massless taut strings that are assumed to represent the centroid lines of muscles along which forces are transmitted to the skeleton (Fig. 2.1). One of the major difficulties lies in the adequate modeling of the interaction between these strings and neighboring obstacle surfaces such as bones and tissue, which cause the strings to wrap around them. This difficulty arises because biologically accurate wrapping surfaces have compex shapes, and a single muscle may wrap around many obstacles. This thesis is dedicated to the problem of computing musculotendon paths across wrapping surfaces, which is well known as the muscle wrapping problem.

The solution of the muscle wrapping problem has been subject of intensive research within the last decades, and a variety of methods has been proposed (Sec. 2.4.3). With existing methods, researchers face a trade-off between computational speed and modeling accuracy. Available methods that are computationally fast can only work with not more than two simple surfaces such as spheres and cylinders that do not accurately represent complex biologic wrapping surfaces. Methods that can work with complex surfaces have a significant impact on the computational costs of musculoskeletal models. This work proposes a new method that allows for the efficient computation of musculotendon paths, their lengths, and their rates of length change across a variable number of general wrapping obstacle surfaces.

**Figure 2.1** Schematic representation of the triceps brachii as a volumetric body **(a)** and as a one-dimensional curve that represents the muscle's centroid line along which force is transmitted to the skeleton **(b)**. Wrapping surfaces ensure that the simplified muscle takes a biologically accurate path such that the computer model renders meaningful muscle path length, rate of length change, and moment arms.




## 2.1 Motivation

The validity of computed muscle forces, moment arms, and internal body loads as well as the overall computational speed of the musculoskeletal model depends significantly on the method used to compute muscle paths. A good muscle wrapping method should fulfill the following three requirements, which have motivated the work presented in this thesis.

- **Generality.** Biologically accurate wrapping surfaces are complex, and a single muscle path may wrap around many obstacles. A general wrapping method should be able to compute muscle paths across multiple wrapping surfaces that accurately represent their biologic counterparts, e.g., surface patches fitted to measured bone geometry. This is necessary to recreate experimentally measured moment arms, muscle path length, and rate of length change over a wide range of joint motion. A general method should furthermore be able to work with both parametric and implicit surface representations to allow for a large assortment of surfaces.
- **Efficiency.** The human musculoskeletal system consists of hundreds of muscles, and a single muscle may wrap around multiple obstacle surfaces. Musculoskeletal models typically contain a large number of muscles, whose individual paths need to be computed continuously during simulation. A good wrapping algorithm should have a small computational footprint to allow for the evaluation of musculoskeletal models on a desktop computer. Computational efficiency is of particular importance

for models that are used in applications involving optimization such as optimal control.

- **Continuity.** Computed muscle paths should slide and evolve continuously during motion to ensure continuous time histories of muscle path length and rate of length change. Continuous behavior is desirable because it allows variable step-size integrators to perform large time steps. In addition, a continuous wrapping method should allow for simulating path lift-off or touchdown. Though such events are rare in biological systems due to connective tissue and little relative motion between muscles and wrapping surfaces, they can occur in musculoskeletal simulations when idealizing muscles by thin lines.

The next section describes the method's underlying assumptions, which are made to fulfill the aforementioned criteria.

## 2.2 Assumptions

This section summarizes the method's underlying assumptions, which are essential for the remaining part of this thesis.

- **Muscle dimension.** Muscles are assumed to be thin one-dimensional lines. Each line represents the muscle's centroid line (Fig. 2.1), sometimes called line of action, along which muscle force is concentrated and transmitted to the skeleton. This modeling assumption applies well to thin muscles, tendons, and ligaments in particular, but requires special care when modeling thick muscles.
- **Muscle mass.** It is assumed that muscles are massless and that muscle mass is lumped with body segments. This assumption is feasible for $(i)$ muscles that perform little motion relative to the skeleton and $(ii)$ muscles that do not undergo large deformations. For such muscles, the centers of mass can be assumed to remain at the same positions relative to the skeleton, and the moments of inertia can be assumed constant. This assumption applies well to thin muscles, tendons, and ligaments but it may introduce errors when simulating volumetric muscles.
- **Muscle friction.** Friction between muscles and wrapping surfaces is neglected, thus no tangential forces are transmitted from muscles to wrapping surfaces and vice versa. Therefore, muscles are assumed to have uniform tension. This assumption is widely accepted because biologic systems are well-lubricated to avoid unnecessary energy losses.

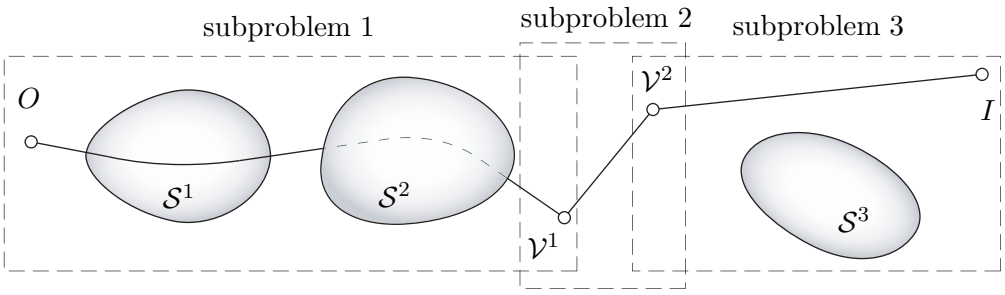
As a result of the above simplifications, the problem of computing a musculotendon's path is reduced to one of finding the locally shortest path between muscle origin and

insertion, while the muscle wraps over a variable number of general obstacle surfaces. The neglection of muscle mass and friction makes the problem a purely geometric one. Thereby its solution depends only on the instantaneous configuration of the wrapping obstacles and on the positions of path origin and insertion. This thesis is dedicated to the efficient solution of the shortest-path problem against the backdrop of modeling musculotendon paths using length-minimizing curves. In the following, some other important side aspects are discussed.

- **Via points.** The computation of muscle paths may not only require wrapping surfaces, but also via points. Via points are typically used to simplify the path-computation problem, or when wrapping surfaces appear to be disproportionately complex. Figure 2.2 exemplarily illustrates a generic shortest-path problem by means of three wrapping surfaces $\mathcal{S}^1$, $\mathcal{S}^2$, and $\mathcal{S}^3$, and two via points $\mathcal{V}^1$ and $\mathcal{V}^2$, where $O$ and $I$ are path origin and insertion, respectively.

  Via points do not complicate the overall nature of the shortest path problem. Any shortest path that contains via points can be regarded as a concatenation of shortest paths without via points. For instance, the path shown in Fig. 2.2 involves two via points, but the shortest path problem can be decomposed into three subproblems without via points. Here, the first subproblem consists in computing segment $O\mathcal{V}^1$, where the latter two consist in computing $\mathcal{V}^1\mathcal{V}^2$, and $\mathcal{V}^2 I$, respectively. The latter two are easily computed as they are straight lines, while segment $O\mathcal{V}^1$ corresponds to the type of problem that is addressed in this thesis, with $O$ being the origin point and $\mathcal{V}^1$ being the insertion point.

**Figure 2.2** Decomposition of a shortest-path problem with two via points into three subproblems without via points. The first subproblem consists in computing segment $O\mathcal{V}^1$, where the latter two consist in computing segments $\mathcal{V}^1\mathcal{V}^2$ and $\mathcal{V}^2 I$, respectively.




  Since via points do not complicate the overall problem, they are not discussed in more detail in this work.

- **Order of surfaces.** Without loss of generality it is assumed that all wrapping obstacles are arranged in a fixed order from origin to insertion, i.e., the surface $\mathcal{S}^1$ is closest to the origin point $O$. The term closest is used here in the sense that for an observer traveling along the shortest path from $O$ to $I$, $\mathcal{S}^1$ is the first surface

the observer walks on. This assumption is motivated by the fact that wrapping obstacles in biologic systems retain a constant order.

- **Path lift-off and touchdown.** In a general shortest-path problem, obstacle surfaces may not be in contact with the path in some situations (Fig. 2.2). Path lift-off and touchdown is rare in biological systems due to connective tissue, but the simplification of muscles to thin lines may cause the appearance of such contact events during simulation. It is assumed that any surface can make and break contact with the path. Surfaces that do not touch the path retain their order with respect to the other surfaces.
- **Solution uniqueness.** In general, there exist several options to span a length-minimizing path between two points. Given the example configuration shown in Fig. 2.2, for instance, surface $\mathcal{S}^3$ is not in contact in the shown situation, but there may also be a feasible solution in which the path segment between $\mathcal{V}^2$ and $I$ wraps across $\mathcal{S}^3$. Both solutions yield a different path length with the latter being greater than the former. The strategy followed in this thesis, however, does not consist in finding the absolute shortest path, but in finding the locally shortest path in the proximity of an estimated configuration that is assumed to be sufficiently close to the sought solution. Given the background of a biological system, it is roughly known on which sides muscles wrap over bones. Therefore estimated configurations are assumed to be known.
- **Temporal coherence.** It is assumed that wrapping obstacles move continuously in time and thus that any configuration of obstacles and muscle paths is coherent to its previous one. The temporal coherence supports the use of an iterative gradient-based method that takes the solution of a previous time step as an initial guess for the current time step. This feedback is essential for the method presented in this thesis.

The next section explains the methodological approach to computing shortest paths and presents an overview over the remaining part of the thesis.

## 2.3 Solution approach

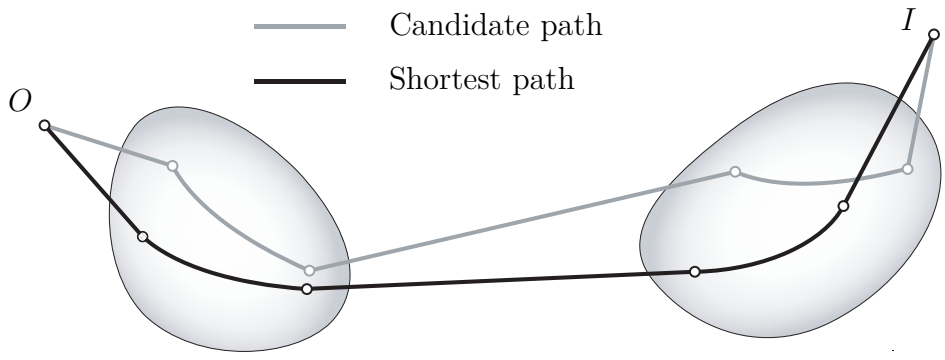
With a view to the previously stated marginal conditions and assumptions, this section provides a methodological overview about the thesis. It explains the general solution approach for computing shortest paths and illustrates it by means of a simple introductory example.

A complete solution of the shortest-path problem for musculoskeletal simulation requires the computation of the following quantities: the shortest path that represents the muscle,

the path length, the path rate of length change, and the forces applied to the wrapping obstacles. As mentioned before, the muscle-path computation does only require knowledge of the instantaneous configuration of the wrapping obstacles and the positions of path origin and insertion. The computation of the path's rate of length change additionally requires knowledge of the spatial velocities of all obstacle surfaces, i.e., their linear and angular velocities, and also the linear velocities of path origin and insertion. Wrapping obstacles as well as origin and insertion points are assumed to be rigidly fixed to the skeleton, thus their positions, orientations, linear and angular velocities are known at any time. As shown later, the computation of the path and its length is by far more complicated than computing its rate of length change and the reaction forces applied to the obstacles. The general procedure for computing the path is illustrated by means of an introductory example in Sec. 2.3.1.

The solution approach to solving the shortest-path problem consists in regarding the total path as a concatenation of shortest-path segments: local geodesic segments on the surfaces, and straight-line segments between the surfaces as well as to path origin and insertion, respectively. At the sought solution configuration, the transitions between adjacent geodesic and straight-line segments are collinear, i.e., the transitions have no kink (Fig. 2.3). These collinearity conditions are used to state a nonlinear path-error constraint equation that serves as a metric to measure how close a candidate path is to the sought solution, where a vanishing path error defines the shortest path. The root of the path-error constraint equation is computed iteratively with a gradient-based root finding method to find the shortest path. Given the positions and orientations of all wrapping surfaces as well as the positions of path origin and insertion are known, the problem of finding the shortest path is reduced to one of finding the parameters of all geodesics such that the path error vanishes.

**Figure 2.3** A shortest path as a concatenation of shortest-path segments: geodesic segments on the surfaces and straight-line segments between the geodesics and to origin $O$ and insertion $I$. At the solution configuration, the transitions between all segments are collinear, i.e., they have no kink. The figure shows one candidate path (grey) with noncollinear transitions and the sought solution (black). The solution is found iteratively using a gradient-based root-finding method.



In this work, each geodesic is naturally parameterized by its start point, its direction, and its length. For the path iterations it is necessary to know the gradient of the path error with respect to the geodesic parameters, i.e., the path-error Jacobian. It allows for the targeted iterative correction of the geodesic parameters such that the path error decrements. In order to obtain the path-error Jacobian, four natural, i.e., independent, variations of the parameters of each geodesic are introduced. By formulating the Jacobian with respect to these four variations, the matrix is independent of the representation of the surfaces, i.e., parametric or implicit. Each path iteration step yields a set of four natural geodesic corrections per geodesic, which are then mapped back to a set of geodesic parameters that correspond to the chosen surface representation. Thereby, the shortest-path problem can be solved very efficiently by applying a gradient-based root-finding method, e.g., Newton's method, with an explicit banded Jacobian. The main idea of this solution approach is illustrated in the following introductory example.

### 2.3.1 Introductory example

This introductory example illustrates the underlying idea of the root-finding method for computing shortest paths by means of a simple two-dimensional setup. Here, the shortest path across a unit circle is sought. For the sake of simplicity, the path's origin point $O$ is located on the circle and its insertion point $I$ is located at a distance of two times the radius from the circle's origin. Accordingly, the shortest path consists of one geodesic segment $\gamma$ with start point $O$ and end point $Q$ on the circle, and one straight-line segment $\sigma$ connecting points $Q$ and $I$ (Fig. 2.4). The solution approach consists in computing the position of $Q$ on the circle such that the geodesic segment connects collinearly with the straight-line segment.

Let the circle be parameterized by

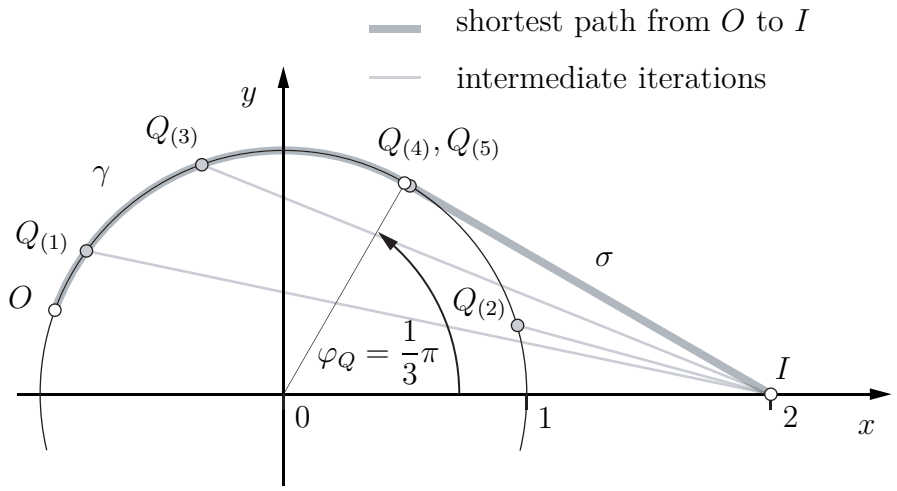
$$\mathbf{x}(\varphi) = \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix} \tag{2.1}$$

with respect to the introduced $xy$ coordinate system, where $\varphi \in [0, 2\pi[$ is the opening angle with respect to the horizontal $x$ axis (Fig. 2.4). Let the position of point $O$ be given by $\varphi_O = 8/9\pi$ and let the position of point $Q$ be parameterized by $\varphi_Q$. Accordingly, the geodesic segment has the length $\ell_\gamma = \|8/9\pi - \varphi_Q\|$. Both segments $\gamma$ and $\sigma$ connect collinearly if the circle's normal vector at $Q$ is perpendicular to the vector from $I$ to $Q$. This collinearity condition can be cast into the following constraint equation

$$\varepsilon(\varphi_Q) = \begin{bmatrix} \cos\varphi_Q \\ \sin\varphi_Q \end{bmatrix} \cdot \begin{bmatrix} \cos\varphi_Q - 2 \\ \sin\varphi_Q \end{bmatrix} = 1 - 2\,\cos\varphi_Q = 0\,, \tag{2.2}$$

whose root defines the shortest path. The quantity $\varepsilon$ measures the deviation from collinearity at the transition between the segments and is therefore termed path error. It is straightforward to compute analytically that the path error vanishes for $\varphi_Q = \pm 1/3\pi$, i.e., Eqn. 2.2 has roots at these points. Suppose here that $\varphi_Q = 1/3\pi$ is the target solution which is sufficiently close to an estimated solution configuration, i.e., a configuration in which the path wraps over the upper half of the circle instead of the lower half of the circle (Fig. 2.4). In the following, the target solution $\varphi_Q = 1/3\pi$ shall be computed iteratively using Newton's method with an explicit path-error Jacobian.

**Figure 2.4** Iterative computation of the shortest path between $O$ and $I$ by finding the position of the transition point $Q$ such that both path segments connect collinearly. The subscript numbers denote iteration steps, where $Q_{(1)}$ is the initial guess and $Q_{(5)}$ is the fifth iteration.




In the general case it is necessary to apply a numerical root-finding method, such as Newton's method, to zero the path error. According to Newton's method, the root of Eqn. 2.2 is solved iteratively by applying the rule

$$\varphi_{Q,(k+1)} = \varphi_{Q,(k)} - \left(\frac{\mathrm{d}\varepsilon}{\mathrm{d}\varphi_Q}\right)^{-1}_{(k)} \varepsilon_{(k)}\,, \tag{2.3}$$

until $\varepsilon$ is sufficiently close to zero. The subscript $k$ in parentheses indicates the current iteration step and $\mathrm{d}\varepsilon/\mathrm{d}\varphi_Q$ is the path-error Jacobian, i.e., the gradient of $\varepsilon$ with respect to $\varphi_Q$. It is given by

$$\frac{\mathrm{d}\varepsilon}{\mathrm{d}\varphi_Q} = 2\,\sin\varphi_Q\,. \tag{2.4}$$

The iteration rule in Eqn. 2.3 requires an initial guess for $k = 1$. This guess corresponds to the estimated solution configuration and should thereby be sufficiently close to the

sought, i.e., unknown, target solution such that Newton's method converges successfully to this solution. In this example, $\varphi_{Q,(1)} = 4/5\pi$ was chosen, displayed as $Q_{(1)}$ in Fig. 2.4. The following Tab. 2.1 shows the numerical values for $\varphi_Q$, $\varepsilon$, and $\mathrm{d}\varepsilon/\mathrm{d}\varphi_Q$ during seven iteration steps, where the first five iteration steps are also illustrated in Fig. 2.4.

**Table 2.1** Iteration results for the given example. The explicit solution $\varphi_Q = 1/3\pi$ is found with high accuracy at the seventh iteration using Newton's method.

| iterations $k$ | path parameter $\varphi_Q/\pi$ | path error $\varepsilon$ | path-error Jacobian $\mathrm{d}\varepsilon/\mathrm{d}\varphi_Q$ |
|---|---|---|---|
| 1 | 0.800000000000000 | 2.618033988749895 | 1.175570504584947 |
| 2 | 0.091113465560706 | −0.918623772225352 | 0.564697105227005 |
| 3 | 0.608925626586802 | 1.671120589930878 | 1.884037460819405 |
| 4 | 0.326588244320030 | −0.036475461311159 | 1.710473214668917 |
| 5 | 0.333376131928492 | 0.000232893232537 | 1.732185247664296 |
| 6 | 0.333333334994133 | 0.000000009037070 | 1.732050812786432 |
| 7 | 0.333333333333333 | 0.000000000000000 | 1.732050807568877 |

The values in Tab. 2.1 reveal some important properties of the gradient-based path iterations. First, the iteration history of the path error $\varepsilon$ shows quadratic convergence near the solution, as expected by using Newton's method. This allows for the computation of high-precision solutions in a few iterations given a sufficiently close initial guess. Second, there is no guarantee that the found solution is the absolute shortest path. A vanishing path error means that the found solution is locally length-minimizing with respect to neighboring curves. In this example, the path-error constraint Eqn. 2.2 has two roots, i.e., one at $\varphi_Q = 1/3\pi$ and another one at $\varphi_Q = -1/3\pi$. The former solution corresponds to a shorter path that wraps over the upper half of the circle while the latter solution corresponds to a longer path that wraps over the lower half of the circle. It is therefore crucial to provide an estimated solution configuration that is sufficiently close to the sought solution.

## 2.4 State of the art

This section contains a review of the literature that is related to this thesis. It is split into three major parts. (*i*) Section 2.4.1 reviews the emergence of the mathematical foundations that are used in this work, with a focus on the history of geodesics. (*ii*) Section 2.4.2 contains an overview over the modeling of musculoskeletal systems and illuminates the emergence of muscle models. (*iii*) Sec. 2.4.3 builds upon the first two sections and summarizes the state of the art of muscle wrapping methods. Note that the

amount of literature available is by far too extensive to be reviewed completely within the scope of this thesis. Therefore, only a selected excerpt is given.

### 2.4.1 Geodesic curves

Geodesic curves have been subject of intense research for some of the most eminent scientists between the late seventeenth century and the late nineteenth century. This section aims to illuminate the essential findings during this time with a focus on the theory that is used in this work. The information contained in this section originates from the following three thorough works: 'Bemerkungen zur Geschichte der geodätischen Linien' by Paul Stäckel (1893) [1], 'Übersicht über die Entwicklung der Theorie der geodätischen Linien seit Gauß' by Paul Sager (1903) [2], and 'Die Geschichte der Differentialgeometrie von Gauß bis Riemann (1828-1868)' by Karin Reich (1973) [3].

The emergence of geodesic curves can be traced back to the year 1697, when Johann Bernoulli posed the problem of finding the shortest curve between two given points on a convex surface to the scientific community. He himself found the fundamental property of such length-minimizing curves, termed geodesic lines or geodesics for short: the normal vector of a geodesic curve is parallel to the normal vector of the surface at any point. Bernoulli asked, among others, Leonhard Euler to find a solution for this problem. In the following century, Euler was the first to obtain the differential equation of geodesics on implicit surfaces represented by an equation of the kind $f(x, y, z) = 0$ with $x$, $y$, and $z$ being cartesian coordinates. Subsequently, Joseph-Louis Lagrange found that the necessary condition formulated by Bernoulli is always fulfilled by curves that are the solution of Euler's differential equation.

The development of the theory of geodesic curves at the beginning of the nineteenth century was substantially propelled by Carl Friedrich Gauß, who worked intensively on cartography and geodesy at this time. Gauß established the relation between geodesic curves and fundamental intrinsic properties of surfaces, such as arc length and curvature. Among other things, he found that the fundamental property of geodesics results directly from the first variation of arc length (Cha. 5), from which he also derived the differential equation of geodesics on parametric surfaces described by $x = x(u, v)$, $y = y(u, v)$, $z = z(u, v)$, where $u$ and $v$ are surface parameters. Gauß was also concerned with investigating the relation between neighboring geodesics that have the same length but different initial conditions, in particular, a different start-point position or a different initial direction. In this setting, he introduced two fundamental lemmata. ($i$) Let a flock of geodesics of identical length be given and let each geodesic originate at the same start point but with a different initial direction. Then the curve connecting the geodesics' end points is perpendicular to all geodesics. These curves are often termed geodesic circles. ($ii$) Let

an arbitrary curve on a surface be given and let geodesics of identical length originate orthogonally at any point of the arbitrary curve in direction of the same side of the curve. Then the curve connecting the end points of the geodesics is perpendicular to the geodesics themselves. Gauß thereby introduced the foundations of two special coordinate systems on curved surfaces. The first one is known as the geodesic polar coordinate system and the second one is known as the geodesic parallel coordinate system. Both systems play a decisive role for the natural geodesic variations described in Sec. 4.3.

Carl Gustav Jacob Jacobi related geodesic curves to the motion of a freely moving point mass on curved surfaces. The trajectory of such a point mass, once given an initial velocity, is a geodesic curve. This analogy is also used in the present thesis to derive the geodesic differential equations (Sec. 3.3.3). Jacobi could thereby apply the principles of mechanics to the theory of geodesic curves, which enabled him to integrate the geodesic equations on an ellipsoid using elliptic coordinates. Moreover, Jacobi raised a key question regarding the global optimality of geodesics. He asked which conditions must be fulfilled such that a geodesic line is indeed the absolute shortest line between two points. For instance there exist two geodesics on a sphere between two arbitrary points, where in general one is shorter than the other. In the special case, when both points have the absolute distance of two times the radius, there even exists an infinite number of geodesics with equal lengths between these points. Jacobi's findings led to a more rigorous definition of geodesics than the minimal-length definition introduced originally by Bernoulli. He defined a geodesic as a curve whose normal is always parallel to the surface normal. This definition has become the standard definition for geodesics and is therefore used in this thesis.

### 2.4.2 Development of muscle models

Researchers have been interested in understanding the function of human and animal musculoskeletal systems for far more than one and a half century. This section reviews the advancement of biomechanics since the mid-nineteenth century and highlights an excerpt of the important scientists and works in this field.

The mid-nineteenth century can roughly be viewed as the beginning of biophysics, the basis of today's computational biomechanics. During this time, several well-known scientists were investigating the function of biologic muscle. As an example, Hermann von Helmholtz [4] investigated the metabolism of frog leg muscles in 1845, showing that a muscle's force generation is the result of chemical processes within the muscle itself, which had only been assumed before by physiologists. Though at this time he lacked more adequate measuring devices to study biologic muscle more accurately, his investigations opened the doors to considering muscles as mechanical actuators that generate mechanical work and heat at a certain metabolic cost. About twenty years later, in 1864, Rudolf

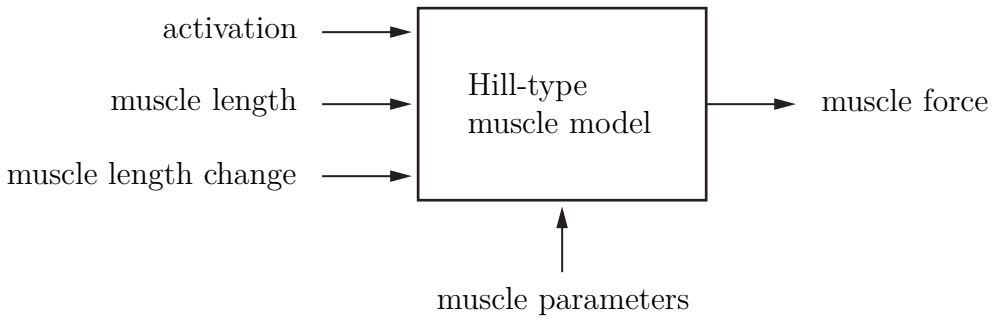
Heidenhain [5] built upon the previous scientific observations and designed experimental setups to relate a muscle's heat generation with its force generation and its metabolism. Among other things he found that a muscle that is inhibited to contract generates more heat than a muscle that is allowed to contract while lifting a weight. He thus concluded that a muscle's heat generation is not the immediate result of mechanical work, which has been a step forward in direction of a better understanding of biologic muscle.

Later on, in 1867, Adolf Fick [6] investigated the amount of mechanical work a muscle can generate. He studied the mechanical work of a frog's gastrocnemius when (*i*) the muscle was initially held at constant length until full tetanus was developed (isometric contraction), and (*ii*) one of its ends was released to lift a weight (isotonic contraction). By showing that a muscle's maximum work can be increased by decreasing its load during contraction, he showed indirectly that a muscle's maximum force depends on its current length. In fact, Fick considered muscles as purely elastic bodies that change their elasticity as the result of stimulus. As a result of this muscle elasticity, he assumed that the maximum force a muscle can generate depends solely on its current length, which had been investigated later, and proven wrong, by Archibald Hill [7, 8, 9, 10].

Fick's mathematical background and his profound knowledge about geometry enabled him to also contribute to the field of biophysics by unvealing fundamental properties of joint motion. In this setting, his major finding was that the motion of anatomic joints which are formed by two bony surfaces can be uniquely defined by a set of rotations (Fick 1885 [11]). He evinced this fact by showing that (*i*) both surfaces must always stay in contact, and (*ii*) each surface must be able to evolve continously on its counterpart. These discoveries were helpful to identify, from a strictly mathematical viewpoint, the possible axes of rotations of anatomical joints. Fick built upon his fundamental discoveries and elucidated the notion of a muscle's moment arm with respect to a given joint. He thereby aroused one of the key questions of today's biomechanics. That is, the question how individual muscles contribute to motion. His solution approach to the determination of individual muscle forces consisted in (*i*) assuming that muscles can be considered as thin force-transmitting straight lines, and (*ii*) geometrically identifying joint axes and effective moment arms. Remarkably, his simplification of muscles as thin lines is still made often today, and is also one of the fundamental assumptions for the muscle wrapping method presented in this thesis. Fick was also well aware of the muscle redundancy problem, also known as the muscle recruitment problem, and presented a first solution approach based on the assumption that humans recruit muscles in such a way that the total effort is minimized. However, the necessary computations turned out to be far too extensive to be performed manually, thus Fick could not achieve quantitative results on muscle forces and paths at this time.

Further progress in metrology brought by the twentieth century enabled researchers to investigate biologic muscle more accurately than before. The first half of this century was substantially influenced by Archibald Hill [7, 8, 9, 10]. His major contribution was the development of mathematical models that relate muscle force with muscle length and contraction velocity, and also with heat dissipation. In 1922 [8] he introduced the idea of muscle viscosity to explain the relation between muscle force and rate of shortening. This idea implies that a muscle occupies the properties of a viscous damper, which prevents it from exerting force rapidly, causing the amount of force a muscle can produce to decrease with increasing contraction velocity. This was experimentally confirmed by means of an apparatus with which humans could lift weights at various speeds. Two years later, in 1924, Gasser and Hill [12] have found this also with isolated muscles, showing that the force-velocity relationship is an intrinsic property of muscle. Based on his observations, Hill found in his famous paper of 1938 [9] that the force-velocity relationship is of hyperbolic fashion and accurately described by the equation $(F + a)(v + b) = c$, where $F$ is the load applied to the muscle, $v$ is the contraction velocity, and $a, b, c$ are muscle-specific constants. He proposed that muscles consist of an elastic element in series with a damped contractile element – a theory which still serves as the basis of muscle models that are used today [13]. A detailed description of a muscle's force-length relationship is given in Hill 1953 [10], showing that muscles have an optimal length at which their generated force is maximal. Because of his fundamental work in muscle mechanics, Hill is regarded as the originator of macroscopic muscle models, often referred to as Hill-type models (Fig. 2.5).

**Figure 2.5** Schematic of a macroscopic Hill-type muscle model of contraction dynamics. Current muscle length and rate of length change are essential inputs of the model for computing muscle force, given neuronal activation and muscle-specific parameters.




Hill-type muscle models aim to reproduce the behavior of biologic muscle in an input-output manner, as shown in Fig. 2.5. That is, they compute tensional muscle-force given neuronal activation, muscle length, rate of length change, and muscle-specific parameters such as maximum isometric force. Hill-type muscle models are today widely used in many applications because of their accuracy that comes at low computational costs and a straightforward applicability.

The beginning of the second half of the twentieth century was shaped by further advances in imaging technology, which revealed microscopic insights into muscle function. Almost coincidently, Andrew Huxley and Rolf Niedergerke (1954) [14] as well as Hugh Huxley and Jean Hanson (1954) [15] discovered that whole-muscle contraction results from the shortening of individual muscle fibers. Muscle fibers shorten because of the interaction between actin and myosin. The actin filaments slide past the myosin filaments due to alternating attachment and detachment of cross-bridges, which ultimately causes the sarcomeres to shorten. These findings have given birth to the so called sliding filament theory, which is accepted today as the most detailed theory of muscle contraction. The sliding filament theory had been investigated thoroughly and also improved within the second half of the twentieth century [16, 17, 18, 19, 20, 21]. Huxley-type microscopic models describe the contractile mechanism of muscle on the molecular level. They are thus well-suited to study the causes of fundamental properties of muscle contraction such as force-length dependency or muscle fatigue. However, the modeling process requires the identification of a large number of microscopic muscle parameters and probability distribution functions, which may not be obtained straightforwardly. As a result, Huxley-type models entail a considerably larger modeling effort than the classical Hill-type models, which prevents a more widespread use in biomechanics.

### 2.4.3 Muscle paths in musculoskeletal models

The rapid advancement of personal computer performance within the last couple of decades has substantially promoted computational biomechanics. In this setting, researchers have been developing and continuously improving computational models of the musculoskeletal system with the aim to investigate the principles of human and animal locomotion.

Today, musculoskeletal simulations provide a quantitative means to predict internal body loads given a set of experimentally measured body kinematics and ground reaction forces. Therefore, musculoskeletal models are widely used to study pathological gait patterns [22, 23, 24, 25, 26], to predict the contribution of individual muscles to motion such as walking [27, 28, 29, 30, 31], running [32, 33], and shoulder motion [34, 35], to quantify bone-on-bone contact forces and joint loads [36, 37, 38, 39], and for surgical planning [40] such as cardiomyoplasty.

Internal body loads and joint forces depend on muscle forces, and also on the paths of muscles inside the body as they determine the moment arms relative to the actuated joints. To better understand the force-generating capacity of muscles, researchers have studied isolated muscles and developed models of contraction dynamics (Sec. 2.4.2). It is now well-established that a muscle's ability to generate force depends on the path's

current length and its rate of length change, yet the in-vivo measurement of muscle paths and forces is extremely challenging. While it is possible to accurately measure the coordinates of muscle origin and insertion points from cadavers [41, 42], experimental data of muscle paths and forces has only been obtained in few cases using MRI [43, 44] and force transducers attached to tendons [45, 46].

Musculoskeletal models require methods that compute muscle paths, their lengths and their rates of length change to determine muscle forces and the contribution of individual muscles to motion. Muscles commonly wrap around multiple complex anatomical obstacles such as bones and neighboring tissue, thus most muscle paths cannot be represented adequately by straight lines. Therefore, a broad variety of muscle wrapping approaches has been reported in the literature, some of which are discussed in more detail below. Available approaches for simulating muscles range from rather simple polyline approaches to highly detailed finite element approaches. Overall, the choice of an appropriate approach depends mainly on the required level of detail, and also on the available computer performance.

Finite element algorithms such as [47, 48, 49, 50] provide the highest level of detail because they represent muscles as deformable volumetric bodies and incorporate mass distributions as well as material properties. Finite element models are typically used to reveal fundamental properties of individual muscles, such as the human tibialis anterior [51] or the masseter muscle [52]. They are also well-suited to predict pressure distributions of muscles on bones. Researchers working with finite element models of muscles face two major challenges. First, the material properties of biologic muscle are difficult to identify. It has nonlinear and anistropic properties, contains active and passive elements, and contracts non-uniformly. Moreover, muscle fibers are electrically insulated, making the realistic modeling of neuronal activation challenging. Second, finite element modeling of muscles requires the coupling of different scales both in time and in space. The duration of the neuronal action potential on a single fiber is a few milliseconds, but the duration of the resulting tension development is a few hundred milliseconds [48]. Muscles as a whole have the size of multiple centimeters, while muscle fascicles are a few millimeters long, and the accurate modeling of sarcomeres with actin and myosin filaments requires a scale of a few micrometers. As a result of their complexity, finite element models are computationally expensive and the modeling process requires special care.

On the other side of the spectrum, there exist applications such as full-body models with many muscles that require rather simple and efficient methods for predicting musculotendon paths. In this setting, approaches have been developed by which muscles are modeled as massless, taut strings that can slide frictionlessly on bony surfaces. By this assumption, muscles are represented as locally length-minimizing curves, i.e., straight lines and geodesics, that transmit forces to the skeleton while wrapping around geometric ob-

stacle surfaces representing bone and tissue. Shortest-path muscle wrapping approaches can be criticized because (*i*) they make it necessary to lump muscle masses and moments of inertia with body segments, which can cause errors in dynamic simulations [53], and (*ii*) the negligence of muscle thickness may yield incorrect moment arms. Despite their limitations, shortest-path muscle wrapping methods are by far the most widely used methods to compute muscle paths. This is because it is straightforward to use them in combination with Hill-type muscle models to predict muscle forces, and also because they can recreate experimentally measured moment arms [54]. Overall, shortest-path muscle wrapping methods have shown to be sufficiently accurate within the biometric tolerances. For instance, they have been used to simulate the upper limb [55, 54, 56], the lower limb [57, 58, 59], and the shoulder [34, 35].

Curved-line muscle wrapping approaches can roughly be subdivided into two groups: approaches using path or surface discretizations such as [57, 54, 60] and approaches using smooth curves on smooth wrapping surfaces such as [55, 61, 62]. Discretized wrapping surfaces such as bone meshes or slices are usually generated from CT or MRI scans [63, 64, 57] and thus they are well-suited to represent complex biologic wrapping surfaces. The computation of shortest paths across discretized surfaces is computationally efficient because the necessary computations are confined to computing distances and angles between straight-line segments. On the downside, discretized surfaces cause nonsmooth rate of length change during path evolution and wrapping over surface edges. This can introduce discontinuities in muscle force and slow down variable step size integrators during simulation. Researchers have also applied nonlinear optimization to compute discretized shortest muscle paths [56, 65] on continuous surfaces. Here, the approach consists in minimizing the energy of a series of lumped springs, given implicit surface equations as unilateral constraints. This approach allows for using multiple implicit surfaces and provides good approximations of the exact shortest path when the level of path discretization is sufficiently high. However, the computational costs grow significantly with the level of path discretization and the number of surfaces, and the method yields nonsmooth rate of length change. Overall, the discretization of wrapping surfaces and muscle paths remains an unfavorable approach to the muscle wrapping problem.

Smooth curves and surfaces are necessary to avoid the nonsmooth behavior of discretizations. An early smooth wrapping approach was presented by van der Helm [34], who used spheres, cylinders, and ellipsoids for single-object wrapping around the shoulder. To simplify the computation, he approximated surface geodesics by planar curves, which introduces errors in path length, rate of length change, and computed moment arms. Garner and Pandy [55] introduced the Obstacle-Set method which computes the exact shortest (geodesic) path across a maximum of two spheres, cylinders, or a combination of both. This method requires a series of case distinctions and does not generalize to more

than two elementary surfaces. Stavness et al. [66] regarded the total path as a concatenation of straight-line segments between two geodesic segments on each surface, where each geodesic segment is assumed to emanate in the direction of its adjacent straight-line segment. They computed the shortest path across multiple implicit surfaces by iterating the origin-point positions of the geodesic segments such that the two geodesic segments on each surface connect collinearly at their closest points, and the adjacent straight-line segments are tangent to the surface. The main advantage of this approach is the ability to work with complex wrapping surfaces that match their biologic counterparts more accurately than spheres and cylinders. On the downside, this apporach is computationally slow as it relies on finite-differences Jacobians and requires nested loops for finding the closest points on each pair of local geodesic segments. Scholz et al. [67] used a single geodesic segment per surface and formulated the constraints for the shortest path solely at the transitions between the geodesic segments and their adjacent straight-line segments. In that work, each geodesic segment was parameterized by the coordinates of its boundary points on general parametric surfaces, and a system of local path-error constraints was introduced which enforces that the transitions between all geodesic segments and adjacent straight-line segments are collinear. That method allowed for the computation of shortest muscle paths across multiple surfaces by solving a system of nonlinear path-error constraints with an explicit Jacobian. The use of an explicit Jacobian makes that method more efficient than [66], yet it still requires nested loops for computing geodesics between two points on a surface. Overall, previous approaches to smooth-surface wrapping have neglected to address the explicit computation of the rate of length change of the muscle path, which is an important input for computing muscle forces using Hill-type muscle models [10, 17, 68, 13].

In summary, it can be stated that there exists a strong demand for a muscle wrapping method that computes a muscle's shortest path over multiple biologically realistic surfaces in real time, as well as the path's exact rate of length change. This is because the few available approaches that can work with general wrapping surfaces are computationally expensive, while the more efficient approaches are limited in the number and shape of available surfaces. As a result, biomechanists face a trade-off between the computational speed and the accuracy of their models.

## 2.5 Overview

The remaining part of this thesis is structured as follows. The subsequent Cha. 3 briefly establishes the differential-geometric foundations of curves and surfaces. It discusses general spatial curves and introduces fundamental notions such as arc length, curvature, and torsion. Thereafter it addresses the fundamental concepts of spatial surfaces such as the

normal vector, the first and second fundamental forms, and the Gaußian curvature. The major part of Cha. 3 is dedicated to the differential-geometric foundations of geodesics. The first variation of arc length is discussed, from which the fundamental property of geodesics is derived. Then the differential equations of geodesic curves on parametric surfaces are derived. From this it becomes clear that a geodesic curve is uniquely defined by a start point, a direction, and a length. Then the rate of length change of a geodesic segment is derived, which is fundamental to the derivation for the total rate of length change of shortest paths. Chapter 3 closes with explaining the concept of a moving Darboux trihedron, which is used to measure the deviation from collinearity at the boundary points of the geodesics to formulate the path error function.

Chapter 4 addresses the mathematical formulation and solution of the shortest-path problem. First, it introduces the required mathematical quantities. Subsequently, the global path-error constraint function is derived from the collinearity conditions at the transitions between straight-line segments and geodesic segments. Thereafter, given the knowledge about geodesic curves explained in Cha. 3, the four natural variations of a geodesic curve are presented. The effect of these variations on the positions of the geodesic boundary-point trihedra is discussed. The presented derivations cover the two famous lemmata of Gauß and the introduction of variational vector fields along geodesics. Using the relation between geodesic variations and variations of the boundary-point trihedra, the global path-error Jacobian is established, which is required for the iterative path corrections. Each correction step yields a set of four finite natural corrections of the parameters of each geodesic. The subsequent section describes the mapping from the finite natural corrections of the geodesic parameters to a new set of geodesic parameters that define the updated start points, directions, and lengths.

In Cha. 5, an explicit formula for the rate of length change is derived using the knowledge about the first variation of arc length described in Cha. 3. The final formula turns out to be independent from the formulation used to solve for the path, and does only include the unit vectors along the straight-line segments together with the rigid-body velocities of the wrapping surfaces and the velocities of path origin and insertion.

The following Cha. 6 addresses the computation of path lift-off and touchdown. To this end, the so-called signed distance between a straight-line segment and a surface is introduced, which serves as a continuous witness function for detecting contact events. Moreover, the normalization of the signed distance with respect to the curvature of surfaces is discussed.

Chapter 7 contains an extension of the method to implicit surfaces. It provides explicit expressions for all quantities that are introduced in the previous sections by means of a generic parametric surface representation. For instance, formulas for Gaußian curvature and geodesic torsion are given. This section also contains an algorithm for computing the

signed distance between the path and implicit surfaces, as well as an example computation of the signed distance between a generic straight-line segment and a paraboloid.

Chapter 8 shows the results of various simulation benchmarks that were used to study the proposed method. The first benchmark addresses the computation of the signed distance between a paraboloid and a straight-line segment, and plots for path length and rate of length change are given. The second benchmark is a dynamic force-driven simulation in which a single muscle path is wrapped over four (nonsimple) surfaces. In this setting, two of the four surfaces are described by an implicit representation, where the other two are described parametrically. In this benchmark, the quadratic convergence behavior of the method is demonstrated. The third benchmark demonstrates the possibility to simulate muscle paths over surface patches that are fitted to realistic bone geometry. Finally in the fourth benchmark, a single muscle is wrapped over a variable number of cylinders to measure the computational costs of the method with respect to the number of surfaces. The benchmark shows that the method's computational costs grow linearly with the number of wrapping obstacles, that it is possible to simulate muscle paths over hundreds of surfaces in real time, and that parametric and implicit surfaces work similarly fast.

Finally, Cha. 9 contains a brief summary of the results achieved by this work, discusses the limitations, and suggests possible future directions of research.

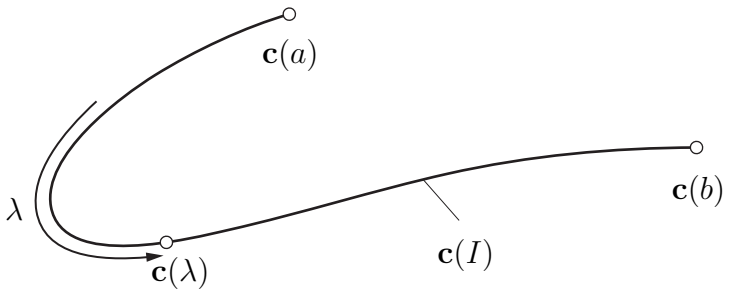
# 3 Differential geometry of curves and surfaces

This chapter briefly describes the fundamental differential-geometric properties of spatial curves and surfaces. In particular, this chapter discusses the properties of geodesics, which are locally length-minimizing curves on surfaces.

## 3.1 Spatial parametric curves

Let $\mathbf{c} : I \mapsto \mathbb{R}^3$, where $I = [a, b] \subset \mathbb{R}$ is a closed interval. The image $\mathbf{c}(I)$ is called curve, and the function $\mathbf{c}(\lambda)$ is called curve parameterization, where $I$ is the parameter domain and $\lambda \in I$ is the curve parameter. The parameterization $\mathbf{c}(\lambda)$ is differentiable if its scalar component functions $c_x(\lambda)$, $c_y(\lambda)$, and $c_z(\lambda)$ are differentiable and it is regular if the derivative $\mathrm{d}\mathbf{c}/\mathrm{d}\lambda$ is nonzero for all $\lambda$.

**Figure 3.1** Image $\mathbf{c}(I)$ with $I = [a, b]$.




In this work it is assumed that any curve is regular and at least two-times differentiable.

### 3.1.1 Arc length of a curve

Suppose that $\mathbf{c}(\lambda)$ is a regular and differentiable parameterization of a curve $\gamma$. An arc-length differential $\mathrm{d}s$ of $\gamma$ can be expressed in terms of the differential $\mathrm{d}\mathbf{c}$ as

$$\mathrm{d}s = \sqrt{\mathrm{d}\mathbf{c} \cdot \mathrm{d}\mathbf{c}}\,, \tag{3.1}$$

where $\cdot$ is the scalar-product operator. This equation directly yields

$$\frac{\mathrm{d}s}{\mathrm{d}\lambda} = \sqrt{\frac{\mathrm{d}\mathbf{c} \cdot \mathrm{d}\mathbf{c}}{\mathrm{d}\lambda^2}}, \tag{3.2}$$

where the derivative $\mathrm{d}s/\mathrm{d}\lambda$ if often called curve speed. Under the assumption that the arc length $s$ is measured from $\lambda = 0$ it holds for the total arc length $s(\lambda)$

$$s(\lambda) = \int_0^{\lambda} \| \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}\bar{\lambda}} \| \, \mathrm{d}\bar{\lambda} \,. \tag{3.3}$$

To avoid the integration of the above Eqn. 3.3 it is favorable to work with unit-speed curves, i.e., curves where $\mathrm{d}\lambda = \mathrm{d}s$ and thus $\lambda = s$. It can be shown that any curve can be reparameterized such that it is a unit-speed curve [69]. Therefore the next sections are build upon the assumption that any curve has unit speed.

### 3.1.2 Frenet-Serret formulas

The notion of the arc length $s$ of a curve allows for introducing the unit-tangent vector along a curve as

$$\mathbf{t} := \mathbf{c}' \,, \quad (\cdot)' := \frac{\mathrm{d}}{\mathrm{d}s}(\cdot) \,, \tag{3.4}$$

where $'$ denotes a derivative with respect $s$. The vector $\mathbf{t}$ has unit length because

$$\mathbf{t} \cdot \mathbf{t} = \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}s} \cdot \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}s} = 1 \,. \tag{3.5}$$

Differentiating the above equation with respect to $s$ yields

$$2\,\mathbf{t}' \cdot \mathbf{t} = 0 \,, \tag{3.6}$$

which means that $\mathbf{t}'$ is orthogonal to $\mathbf{t}$. The unit normal vector $\mathbf{n}$ of a curve is defined by normalizing $\mathbf{t}'$

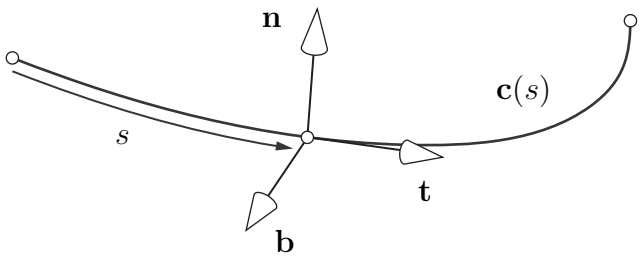
$$\mathbf{n} := \frac{1}{\kappa}\,\mathbf{t}' \,, \tag{3.7}$$

where the scalar quantity $\kappa = \| \mathbf{t}' \|$ is the curvature of the curve. The tangent and normal vectors allow for introducing a third vector, the binormal vector

$$\mathbf{b} := \mathbf{t} \times \mathbf{n} \tag{3.8}$$

that completes the Frenet-Serret frame. Figure 3.2 shows a Frenet-Serret frame along a generic curve.

**Figure 3.2** Frenet-Serret frame of an arc-length parameterized curve with tangent $\mathbf{t}$, normal $\mathbf{n}$, and binormal $\mathbf{b}$.




It is straightforward to show that

$$\mathbf{b}' = -(\mathbf{b} \cdot \mathbf{n}')\,\mathbf{n} = -\tau\,\mathbf{n}\,, \tag{3.9}$$

where the term $\mathbf{b} \cdot \mathbf{n}'$ is called torsion $\tau$ of the curve. With the scalar quantities $\kappa$ and $\tau$ the derivatives $\{\mathbf{t}', \mathbf{n}', \mathbf{b}'\}$ can be written compactly in terms of $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\}$ as

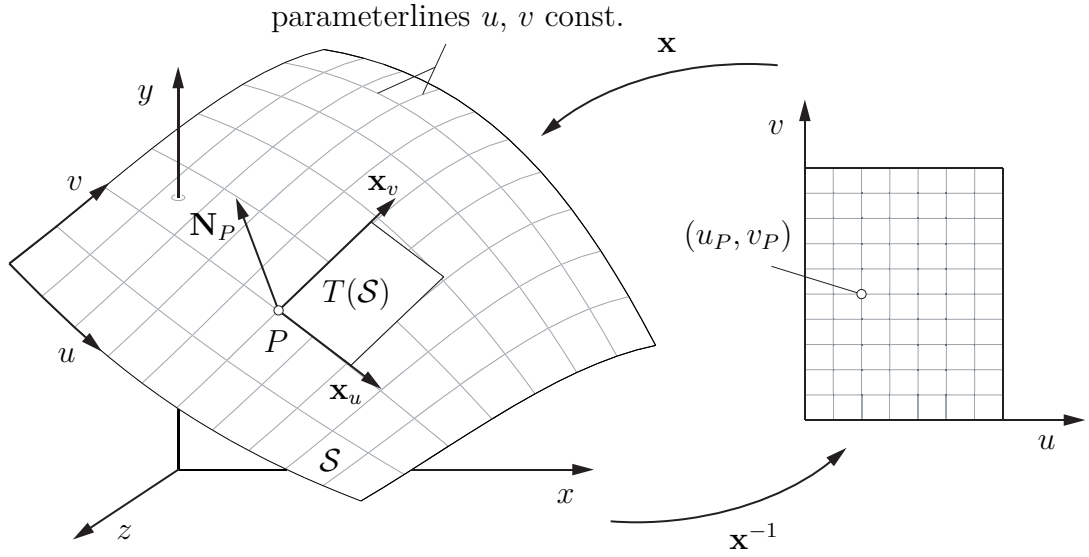
$$\begin{bmatrix} \mathbf{t}' \\ \mathbf{n}' \\ \mathbf{b}' \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{n} \\ \mathbf{b} \end{bmatrix}.$$

The above formulas for $\{\mathbf{t}', \mathbf{n}', \mathbf{b}'\}$ are well-known as the Frenet-Serret formulas in the literature. They completely describe the motion of the frame $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\}$ along a general spatial curve.

## 3.2 Spatial parametric surfaces

A subset $\mathcal{S} \in \mathbb{R}^3$ is a regular surface if for each $P \in \mathcal{S}$ there exists a neighborhood $V$ in $\mathbb{R}^3$ and a map $\mathbf{x} : U \mapsto V \cap \mathcal{S}$ of an open set $U \subset \mathbb{R}^2$ onto $V \cap \mathcal{S} \subset \mathbb{R}^3$ such that $\mathbf{x}$ is a differentiable homeomorphism and the differential $\mathrm{d}\mathbf{x} : \mathbb{R}^2 \mapsto \mathbb{R}^3$ is one-to-one [70]. The open set $U$ is the parameter domain of the parameterization $\mathbf{x}$ and $(u, v) \in U$ are the respective surface parameters.

**Figure 3.3** Parametric surface as a mapping $\mathbf{x}(u, v)$ of two surface coordinates $u$ and $v$ to three Euclidian coordinates $x$, $y$, and $z$. The tangent plane $T(\mathcal{S})$ of a parametric surface is spanned by the partial derivatives $\mathbf{x}_u$ and $\mathbf{x}_v$ of $\mathbf{x}$ with respect to $u$ and $v$, respectively.




Suppose that the parameterization $\mathbf{x}$ is differentiable and regular, then the tangential vectors $\mathbf{x}_u = \partial\mathbf{x}/\partial u$ and $\mathbf{x}_v = \partial\mathbf{x}/\partial v$ span the tangent space of $\mathcal{S}$. Note that the common subscript notation is used here to indicate a partial derivative. With the vectors $\mathbf{x}_u$ and $\mathbf{x}_v$ the normal vector of a parametric surface $\mathcal{S}$ can be defined as

$$\mathbf{N} := \frac{\mathbf{x}_u \times \mathbf{x}_v}{\| \mathbf{x}_u \times \mathbf{x}_v \|}, \quad \mathbf{x}_{(\cdot)} := \frac{\partial \mathbf{x}}{\partial (\cdot)}, \tag{3.10}$$

where $\times$ is the cross-product operator. Without loss of generality it is assumed that the normal vector $\mathbf{N}$ points outside the surface.

### 3.2.1 First fundamental form

Suppose that $\mathbf{x}(u, v)$ is a parameterization of a surface $\mathcal{S}$ and suppose that $\mathbf{x}(u(s), v(s))$ is an arc-length parameterization of a curve on $\mathcal{S}$. For an arc-length element $\mathrm{d}s$ of the curve it holds

$$\mathrm{d}s^2 = \mathrm{d}\mathbf{x} \cdot \mathrm{d}\mathbf{x}, \tag{3.11}$$

$$\mathrm{d}s^2 = (\mathbf{x}_u \,\mathrm{d}u + \mathbf{x}_v \,\mathrm{d}v) \cdot (\mathbf{x}_u \,\mathrm{d}u + \mathbf{x}_v \,\mathrm{d}v), \tag{3.12}$$

$$\mathrm{d}s^2 = \mathrm{E}\,\mathrm{d}u^2 + 2\,\mathrm{F}\,\mathrm{d}u\,\mathrm{d}v + \mathrm{G}\,\mathrm{d}v^2, \tag{3.13}$$

where

$$\mathrm{E} := \mathbf{x}_u \cdot \mathbf{x}_u \,, \tag{3.14}$$

$$\mathrm{F} := \mathbf{x}_u \cdot \mathbf{x}_v \,, \tag{3.15}$$

$$\mathrm{G} := \mathbf{x}_v \cdot \mathbf{x}_v \,. \tag{3.16}$$

The right-hand side of Eqn. 3.13 is known as the first fundamental form of a parametric surface, with the characteristic coefficients E, F, and G defined by Eqn. 3.14, Eqn. 3.15, and Eqn. 3.16, respectively. The first fundamental form allows for the computation of intrinsic geometric quantities on the surface such as the length of a curve, angles between tangent vectors, and areas [70]. Its coefficients contain elementary information about the chosen parameterization and are important for later considerations. For instance, if $\mathrm{F} = \mathbf{x}_u \cdot \mathbf{x}_v = 0$, the chosen parameterization is orthogonal.

Rewriting Eqn. 3.13 using matrix notation yields the metric tensor of the surface

$$\mathrm{d}s^2 = \begin{bmatrix} \mathrm{d}u & \mathrm{d}v \end{bmatrix} \underbrace{\begin{bmatrix} \mathrm{E} & \mathrm{F} \\ \mathrm{F} & \mathrm{G} \end{bmatrix}}_{\text{metric tensor}} \begin{bmatrix} \mathrm{d}u \\ \mathrm{d}v \end{bmatrix} . \tag{3.17}$$

The metric tensor is positive definite since $\mathrm{d}s^2$ is always positive and thus it holds [69]

$$\mathrm{E}\,\mathrm{G} - \mathrm{F}^2 > 0 \,. \tag{3.18}$$

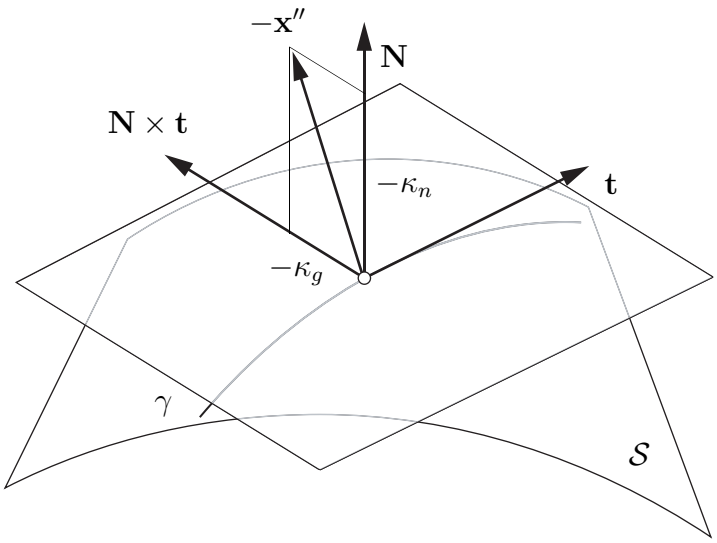
The above inequality constraint is essential for later considerations because the term $\mathrm{E}\,\mathrm{G} - \mathrm{F}^2$ happens to appear in the denominator of some quantities that are introduced in the following sections.

### 3.2.2 Second fundamental form

Suppose that $\mathbf{x}(u(s), v(s))$ is the parameterization of a curve $\gamma$ on a surface $\mathcal{S}$. The derivative of the tangent vector $\mathbf{t}' = \mathbf{x}''$ of $\gamma$ contains information about the curvature of $\mathcal{S}$. The curvature of $\gamma$ contains two components: $(i)$ a normal component $\kappa_n$ and $(ii)$ a tangential or geodesic component $\kappa_g$ (Fig. 3.4). The goal here is to obtain information about the normal curvature $\kappa_n$ of $\mathcal{S}$ in the direction of $\gamma$. This is done by differentiating the expression $\mathbf{t} \cdot \mathbf{N} = 0$ with respect to arc length, yielding

$$\mathbf{t}' \cdot \mathbf{N} + \mathbf{t} \cdot \mathbf{N}' = 0 \,. \tag{3.19}$$

**Figure 3.4** The total curvature of generic curve $\gamma$ on a surface $\mathcal{S}$ can be split into two components: normal curvature $\kappa_n$ and tangential or geodesic curvature $\kappa_g$. The normal curvature is computed by projecting $\mathbf{x}''$ onto $\mathbf{N}$, while the geodesic curvature is obtained by projecting $\mathbf{x}''$ onto $\mathbf{N} \times \mathbf{t}$.




The first summand in the above Eqn. 3.19 is equal to the sought quantity $\kappa_n$, where the second summand still needs to be evaluated. Inserting $\mathbf{t} = \mathrm{d}\mathbf{x}/\mathrm{d}s$ and $\mathbf{N}' = \mathrm{d}\mathbf{N}/\mathrm{d}s$ into Eqn. 3.19 and solving for $\kappa_n$ gives

$$\kappa_n := -\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} \cdot \frac{\mathrm{d}\mathbf{N}}{\mathrm{d}s} \,. \tag{3.20}$$

Inserting $\mathrm{d}\mathbf{N} = \mathbf{N}_u \,\mathrm{d}u + \mathbf{N}_v \,\mathrm{d}v$ and $\mathrm{d}\mathbf{x} = \mathbf{x}_u \,\mathrm{d}u + \mathbf{x}_v \,\mathrm{d}v$ gives [69]

$$\kappa_n = -\frac{(\mathbf{x}_u \cdot \mathbf{N}_u)\,\mathrm{d}u^2 + (\mathbf{x}_u \cdot \mathbf{N}_v + \mathbf{x}_v \cdot \mathbf{N}_u)\,\mathrm{d}u\,\mathrm{d}v + (\mathbf{x}_v \cdot \mathbf{N}_v)\,\mathrm{d}v^2}{\mathrm{d}s^2} \,. \tag{3.21}$$

It is possible to write the above Eqn. 3.21 without the partial derivatives of $\mathbf{N}$ by using the two relationships $\mathbf{x}_u \cdot \mathbf{N} = 0$ and $\mathbf{x}_v \cdot \mathbf{N} = 0$. Differentiating these equations with respect to $u$ and $v$, respectively, yields $\mathbf{x}_u \cdot \mathbf{N} = -\mathbf{x}_{uu} \cdot \mathbf{N}$ and $\mathbf{x}_v \cdot \mathbf{N} = -\mathbf{x}_{vv} \cdot \mathbf{N}$. Inserting these relations as well as Eqn. 3.13 into Eqn. 3.21 simplifies it to

$$\kappa_n = \frac{\mathrm{L}\,\mathrm{d}u^2 + 2\,\mathrm{M}\,\mathrm{d}u\,\mathrm{d}v + \mathrm{N}\,\mathrm{d}v^2}{\mathrm{E}\,\mathrm{d}u^2 + 2\,\mathrm{F}\,\mathrm{d}u\,\mathrm{d}v + \mathrm{G}\,\mathrm{d}v^2} \,, \tag{3.22}$$

where

$$\mathrm{L} := \mathbf{x}_{uu} \cdot \mathbf{N}\,, \tag{3.23}$$
$$\mathrm{M} := \mathbf{x}_{uv} \cdot \mathbf{N}\,, \tag{3.24}$$
$$\mathrm{N} := \mathbf{x}_{vv} \cdot \mathbf{N}\,, \tag{3.25}$$
$$\tag{3.26}$$

are the coefficients of the second fundamental form, which allows for measuring the curvature $\kappa_n$ of a surface in a certain direction of $\gamma$ given by $\mathrm{d}u$ and $\mathrm{d}v$.

### 3.2.3 Gaußian curvature

There are two directions in which the normal curvature $\kappa_n$ has extreme values. The corresponding curvatures $\kappa_1$ and $\kappa_2$ are called the principal curvatures of $\mathcal{S}$. To compute them, parameterize $\zeta = \mathrm{d}v/\mathrm{d}u$ and insert it into Eqn. 3.22

$$\kappa_n(\zeta) = \frac{\mathrm{L} + 2\,\mathrm{M}\,\zeta + \mathrm{N}\,\zeta^2}{\mathrm{E} + 2\,\mathrm{F}\,\zeta + \mathrm{G}\,\zeta^2}\,. \tag{3.27}$$

Then, $\kappa_1$ and $\kappa_2$ are the roots of $\mathrm{d}\kappa_n(\zeta)/\mathrm{d}\zeta$ which can be written as [71]

$$\begin{vmatrix} \mathrm{L} - \kappa_n\,\mathrm{E} & \mathrm{M} - \kappa_n\,\mathrm{F} \\ \mathrm{M} - \kappa_n\,\mathrm{F} & \mathrm{N} - \kappa_n\,\mathrm{G} \end{vmatrix} = 0\,. \tag{3.28}$$

Solving this equation gives

$$\kappa_{1,2} = H \pm \sqrt{H^2 - K}\,, \tag{3.29}$$

where

$$K = \frac{\mathrm{L}\,\mathrm{N} - \mathrm{M}^2}{\mathrm{E}\,\mathrm{G} - \mathrm{F}^2}\,, \tag{3.30}$$
$$H = \frac{\mathrm{L}\,\mathrm{G} - 2\,\mathrm{M}\,\mathrm{F} + \mathrm{N}\,\mathrm{E}}{2\,(\mathrm{E}\,\mathrm{G} - \mathrm{F}^2)}\,. \tag{3.31}$$

The scalar quantity $K = \kappa_1\,\kappa_2$ is the Gaußian curvature and $H = (\kappa_1 + \kappa_2)/2$ is the mean curvature of a surface. Recall here that the term $\mathrm{E}\,\mathrm{G} - \mathrm{F}^2$ that appears in the denominators of $K$ and $H$ is always nonzero because of the inequality in Eqn. 3.18. It shall be noted here that $K$ is an intrinsic surface property, thus it can be written solely

in terms of the first fundamental form [69]. However, it is often convenient to compute the Gaußian curvature using Eqn. 3.30.

## 3.3 Geodesics

Historically, a geodesic is the locally shortest path between two points on a curved surface (Sec. 2.4.1). It is therefore the generalization of a straight line in $\mathbb{R}^3$. This definition is used to derive the fundamental property of geodesics via arc-length variation in the following, referring to [72]. Moreover, this section contains derivations for the rate of length change of geodesics as well as for the geodesic equations.

### 3.3.1 Variation of arc length

Let $\gamma$ be a geodesic given by $\mathbf{x} : [s_0, s_1] \mapsto \mathcal{S}$ on a surface $\mathcal{S}$ and let $P \in \mathcal{S}$ and $Q \in \mathcal{S}$ be the fixed start and end points, respectively. Suppose that $\mathbf{x}(s, \epsilon)$ is a proper variation of $\gamma$ such that $\mathbf{x}(s, 0) = \mathbf{x}(s)$ and $\mathbf{x}_\epsilon(s_0, \epsilon) = \mathbf{x}_\epsilon(s_1, \epsilon) = \mathbf{0}$. Since $\gamma$ is a geodesic, it minimizes the arc length $\ell_\gamma = s_1 - s_0$ between $P$ and $Q$ (Eqn. 3.3). Thus the following condition is fulfilled

$$\gamma : \min_{\epsilon} \quad \ell_\gamma(\epsilon) = \int_{s_0}^{s_1} \| \mathbf{x}'(s, \epsilon) \| \, \mathrm{d}s \, . \tag{3.32}$$

The function $\ell_\gamma(\epsilon)$ must have an absolute minimum at $\epsilon = 0$ because all other curves between $P$ and $Q$ with $\| \epsilon \| > 0$ have a greater arc length. Thus it holds for $\epsilon = 0$

$$\frac{\mathrm{d}\ell_\gamma(0)}{\mathrm{d}\epsilon} = \frac{\mathrm{d}}{\mathrm{d}\epsilon} \int_{s_0}^{s_1} \| \mathbf{x}'(s, 0) \| \, \mathrm{d}s = 0 \, . \tag{3.33}$$

As both variables $s$ and $\epsilon$ are independent, integration and differentiation can be swapped. After differentiation, the above equation becomes

$$\frac{\mathrm{d}\ell_\gamma(0)}{\mathrm{d}\epsilon} = \int_{s_0}^{s_1} \frac{\mathbf{x}'(s, 0) \cdot \mathbf{x}'_\epsilon(s, 0)}{\sqrt{\mathbf{x}'(s, 0) \cdot \mathbf{x}'(s, 0)}} \, \mathrm{d}s = 0 \, . \tag{3.34}$$

After insertion of $\mathbf{x}'(s, 0) \cdot \mathbf{x}'(s, 0) = 1$ and partial integration, the above equation simplifies to

$$\frac{\mathrm{d}\ell_\gamma(0)}{\mathrm{d}\epsilon} = [\mathbf{x}'(s, 0) \cdot \mathbf{x}_\epsilon(s, 0)]_{s_0}^{s_1} - \int_{s_0}^{s_1} \mathbf{x}''(s, 0) \cdot \mathbf{x}_\epsilon(s, 0) \, \mathrm{d}s = 0 \, . \tag{3.35}$$

The left term in the above equation vanishes due to the proper variation conditions and thus the minimum-length condition of $\gamma$ takes the compact form

$$\delta\ell_\gamma = -\int_{s_0}^{s_1} \mathbf{x}''(s,0)\cdot\mathbf{x}_\epsilon(s,0)\,\mathrm{d}s = 0\,, \tag{3.36}$$

where $\delta\ell_\gamma$ is the first variation of arc length. Since the vectors $\mathbf{x}_\epsilon(s,\epsilon)$ are arbitrary vectors lying in the tangent plane $T(\mathcal{S})$ the integral is equal to zero if $\mathbf{x}''(s,0)$ is $(i)$ zero or $(ii)$ normal to $T(\mathcal{S})$. Condition $(i)$ is fulfilled if $\mathbf{x}(s,0)$ describes a straight line whereas condition $(ii)$ means that the normal vector of the geodesic and the normal vector of the surface are parallel. Hence, a geodesic is a curve with zero geodesic curvature.

### 3.3.2 Rate of length change of geodesics

In this work, a geodesic is not considered as a static, infinitely long object on a surface, but as a dynamically moving object with defined boundary points $P$ and $Q$, where $P$ is the start point and $Q$ is the end point. It is therefore not only important to know a geodesic's length, but also to know its rate of length change when its boundary points move on the surface.

The rate of length change of a geodesic segment $\gamma$ results directly from the first variation of arc length. Suppose that $\ell_\gamma$ is the curve's total arc-length, then the first variation $\delta\ell_\gamma$ is (Sec. 3.3.1)

$$\delta\ell_\gamma = \left[\mathbf{x}'(s,\epsilon)\cdot\mathbf{x}_\epsilon(s,\epsilon)\right]_0^{\ell_\gamma} - \int_0^{\ell_\gamma} \mathbf{x}''(s,\epsilon)\cdot\mathbf{x}_\epsilon(s,\epsilon)\,\mathrm{d}s\,, \tag{3.37}$$

In contrast to the considerations in Sec. 3.3.1, consider now only such geodesic variations which themselves are geodesics, i.e., the variations fulfill the minimal-length condition. Under this assumption, the integral in the above Eqn. 3.37 vanishes because $\mathbf{x}_\epsilon$ is tangent to the surface and $\mathbf{x}''$ is normal to the surface according to the fundamental property of geodesics. Consider furthermore only the case $\epsilon = 0$, i.e., variations of a given geodesic between two points $P$ and $Q$. Under these two assumptions, the variation of arc length $\delta\ell_\gamma$ depends only on the variations of the geodesic's end points

$$\delta\ell_\gamma = \mathbf{x}'(\ell_\gamma,0)\cdot\mathbf{x}_\epsilon(\ell_\gamma,0) - \mathbf{x}'(0,0)\cdot\mathbf{x}_\epsilon(0,0)\,. \tag{3.38}$$

For the special case $\epsilon = 0$ it holds

$$\mathbf{x}'(0,0) = \mathbf{t}_P\,, \quad \mathbf{x}'(\ell_\gamma, 0) = \mathbf{t}_Q\,. \tag{3.39}$$

By substituting

$$\mathbf{x}_\epsilon(0,0) = \mathrm{d}\mathbf{x}_P\,, \quad \mathbf{x}_\epsilon(\ell_\gamma, 0) = \mathrm{d}\mathbf{x}_Q\,, \quad \delta\ell_\gamma = \mathrm{d}\ell_\gamma\,, \tag{3.40}$$

the differential $\mathrm{d}\ell_\gamma$ can be written as

$$\mathrm{d}\ell_\gamma = \mathrm{d}\mathbf{x}_Q \cdot \mathbf{t}_Q - \mathrm{d}\mathbf{x}_P \cdot \mathbf{t}_P\,, \tag{3.41}$$

or, after substituting $\mathrm{d}s_P = \mathrm{d}\mathbf{x}_P \cdot \mathbf{t}_P$ and $\mathrm{d}s_Q = \mathrm{d}\mathbf{x}_Q \cdot \mathbf{t}_Q$ it holds

$$\mathrm{d}\ell_\gamma = \mathrm{d}s_Q - \mathrm{d}s_P\,, \tag{3.42}$$

where $\mathrm{d}s_P$ and $\mathrm{d}s_Q$ are arc-length differentials that correspond to a tangential displacement of the boundary points $P$ and $Q$, respectively.

Equation 3.42 and Eqn. 3.41 reveal that the rate of length change of a geodesic depends only on the displacements $\mathrm{d}\mathbf{x}_P$ and $\mathrm{d}\mathbf{x}_Q$ of the end points $P$ and $Q$. More precisely, only the components of $\mathrm{d}\mathbf{x}_P$ and $\mathrm{d}\mathbf{x}_Q$ that are tangent to the geodesic change the curve's length. Vice versa, this means that a displacement of a boundary point in a direction orthogonal to the tangent does not change the curve's length.

### 3.3.3 Geodesic equations

There exists a variety of approaches in the literature to derive the geodesic equations. A thorough derivation would by far overstretch the scope of this thesis. Thus, only a brief and intuitive derivation is given here in order to establish the geodesic equations.

Consider a point mass $m$ that slides freely on a surface $\mathcal{S}$ parameterized by $\mathbf{x}(u, v)$. Freely in this setting means that no forces act on $m$ during the motion, other than the normal force that keeps $m$ on $\mathcal{S}$. If $\mathcal{S}$ was a plane, the motion of $m$ would be a straight line. On a curved surface, $m$ would travel on a general geodesic curve. This fact is used to derive the geodesic equations.

With the given parameterization, the position of $m$ at time $t$ is given by $\mathbf{x} = \mathbf{x}(u(t), v(t))$, while its velocity is given by $\dot{\mathbf{x}} = \dot{\mathbf{x}}(u(t), v(t), \dot{u}(t), \dot{v}(t))$, where the dot notation is used to indicate a time derivative. The parameters $(u, v)$ are the generalized coordinates of

$m$, while the rates $(\dot{u}, \dot{v})$ are the particle's generalized speeds. The motion of $m$ can be expressed by two Euler-Lagrange equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \dot{\mathbf{x}}^2}{\partial \dot{u}}\right) - \frac{\partial \dot{\mathbf{x}}^2}{\partial u} = 0\,, \quad \dot{(\cdot)} := \frac{\mathrm{d}}{\mathrm{d}t}(\cdot)\,, \tag{3.43}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \dot{\mathbf{x}}^2}{\partial \dot{v}}\right) - \frac{\partial \dot{\mathbf{x}}^2}{\partial v} = 0\,, \tag{3.44}$$

where

$$\dot{\mathbf{x}}^2 = \mathrm{E}\,\dot{u}^2 + 2\,\mathrm{F}\,\dot{u}\,\dot{v} + \mathrm{G}\,\dot{v}^2 \tag{3.45}$$

corresponds to the Lagrange function, i.e., the masse's kinetic energy for $m = 2$. Since no tangential forces act on the point mass, its speed is constant and therefore it holds $\mathrm{d}s/\mathrm{d}t = \text{constant}$. Without loss of generality, $\mathrm{d}s = \mathrm{d}t$ can be set and thus the time derivatives in Eqn. 3.43 and Eqn. 3.44 can be replaced by derivatives with respect to arc length. From here on, any geodesic is considered as parameterized by arc length. Transforming Eqn. 3.43 and Eqn. 3.44 into normal form gives

$$u'' + \Gamma^1_{11}\,(u')^2 + 2\,\Gamma^1_{12}\,u'\,v' + \Gamma^1_{22}\,(v')^2 = 0\,, \tag{3.46}$$

$$v'' + \Gamma^2_{11}\,(u')^2 + 2\,\Gamma^2_{12}\,u'\,v' + \Gamma^2_{22}\,(v')^2 = 0\,, \tag{3.47}$$

where the Christoffel symbols in Eqn. 3.46 and Eqn. 3.47 are

$$\Gamma^1_{11} = \frac{\frac{1}{2}\,\mathrm{E}_u\,\mathrm{G} - \mathrm{A}\,\mathrm{F}}{\mathrm{E}\,\mathrm{G} - \mathrm{F}^2}\,, \tag{3.48}$$

$$\Gamma^1_{12} = \frac{\mathrm{C}\,\mathrm{G} - \mathrm{D}\,\mathrm{F}}{\mathrm{E}\,\mathrm{G} - \mathrm{F}^2}\,, \tag{3.49}$$

$$\Gamma^1_{22} = \frac{\mathrm{B}\,\mathrm{G} - \frac{1}{2}\,\mathrm{F}\,\mathrm{G}_v}{\mathrm{E}\,\mathrm{G} - \mathrm{F}^2}\,, \tag{3.50}$$

$$\Gamma^2_{11} = \frac{\frac{1}{2}\,\mathrm{E}_u\,\mathrm{F} - \mathrm{A}\,\mathrm{E}}{\mathrm{F}^2 - \mathrm{E}\,\mathrm{G}}\,, \tag{3.51}$$

$$\Gamma^2_{12} = \frac{\mathrm{C}\,\mathrm{F} - \mathrm{D}\,\mathrm{E}}{\mathrm{F}^2 - \mathrm{E}\,\mathrm{G}}\,, \tag{3.52}$$

$$\Gamma^2_{22} = \frac{\mathrm{B}\,\mathrm{F} - \frac{1}{2}\,\mathrm{E}\,\mathrm{G}_v}{\mathrm{F}^2 - \mathrm{E}\,\mathrm{G}}\,, \tag{3.53}$$

with $\mathrm{A} = \mathbf{x}_{uu}\cdot\mathbf{x}_v$, $\mathrm{B} = \mathbf{x}_u\cdot\mathbf{x}_{vv}$, $\mathrm{C} = \mathbf{x}_u\cdot\mathbf{x}_{uv}$, $\mathrm{D} = \mathbf{x}_{uv}\cdot\mathbf{x}_v$, $\mathrm{E}_u = 2\,\mathbf{x}_{uu}\cdot\mathbf{x}_u$, $\mathrm{G}_v = 2\,\mathbf{x}_{vv}\cdot\mathbf{x}_v$.

The geodesic equations (Eqn. 3.46 and Eqn. 3.47) are $2^{\text{nd}}$ order ODEs. They have a unique solution for a given initial start point $P$ with coordinates $(u_P, v_P)$ and a given initial direction parameterized by $(u'_P, v'_P)$. Note that the magnitudes of $(u'_P, v'_P)$ also define

the geodesic's curve speed. It is essential that $(u'_P, v'_P)$ are chosen such that computed geodesics have unit speed because in that case, the independent integration parameter of the geodesic equations is equal to the arc length of the curve. The unit-speed constraint is written as

$$\| \mathbf{x}' \| = \| \mathbf{x}_u \, u' + \mathbf{x}_v \, v' \| = 1 \, . \tag{3.54}$$

Without loss of generality, the initial value of the length of a geodesic can be assumed to be zero. Thus any geodesic on a parametric surface can be parameterized by the following five parameters

$$\mathbf{q} := \begin{bmatrix} u_P & v_P & u'_P & v'_P & \ell_\gamma \end{bmatrix}^{\mathrm{T}} , \tag{3.55}$$

where $\ell_\gamma$ is the geodesic's total arc length. The first four parameters in $\mathbf{q}$ correspond to the initial values of the geodesic equations, while the fifth parameter corresponds to the upper boundary value for the independent integration parameter (the lower boundary value is assumed to be zero). Note here that the pair $(u'_P, v'_P)$ is not independent because of the unit-speed constraint in Eqn. 3.54.

## 3.4 Darboux trihedron

This section establishes the differential-geometric properties of moving trihedra along curves on surfaces, named Darboux trihedra after Jean Gaston Darboux. The Darboux trihedron is a fundamental concept in this thesis because it is used to formulate the global path error.

Suppose that $\mathbf{x}(u, v)$ parameterizes a surface $\mathcal{S}$ with the normal vector $\mathbf{N}$ and suppose that $\mathbf{x}(u(s), v(s))$ parameterizes a curve $\gamma$ on $\mathcal{S}$ with the tangent vector $\mathbf{t}$. By introducing the binormal vector

$$\mathbf{B} := \mathbf{t} \times \mathbf{N}, \tag{3.56}$$

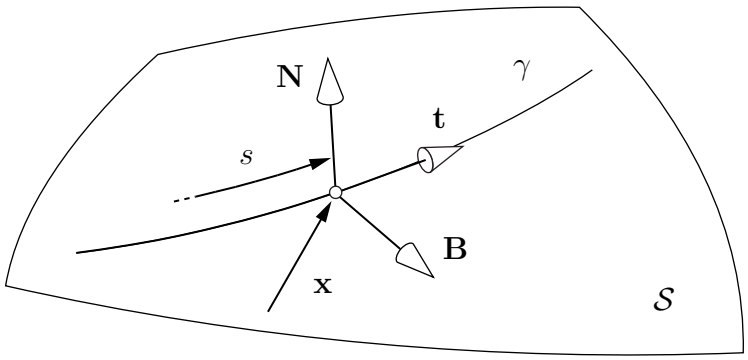
an orthogonal trihedron

$$\mathcal{K} := \{\mathbf{x}, \mathbf{t}, \mathbf{N}, \mathbf{B}\} \tag{3.57}$$

is established along the curve (Fig. 3.5), where $\mathbf{x}$ describes the origin point of $\mathcal{K}$. The vectors $\mathbf{t}$ and $\mathbf{B}$ span the tangent space of $\mathcal{S}$ to which $\mathbf{N}$ is orthogonal. Describing

the motion of $\mathcal{K}$ as a function of arc length requires the computation of the derivatives $\{\mathbf{x}', \mathbf{t}', \mathbf{N}', \mathbf{B}'\}$. Darboux showed that these derivatives can be expressed compactly in the basis $\{\mathbf{t}, \mathbf{N}, \mathbf{B}\}$ using three characteristic scalar quantities, as described next.

**Figure 3.5** Darboux trihedron $\mathcal{K} = \{\mathbf{x}, \mathbf{t}, \mathbf{N}, \mathbf{B}\}$ along a curve $\gamma$ on a surface $\mathcal{S}$. The trihedron consists of the vector $\mathbf{x}$ describing the origin point, the curve's tangent $\mathbf{t}$, the surface's normal $\mathbf{N}$, and the binormal $\mathbf{B} = \mathbf{t} \times \mathbf{N}$.




When $\mathcal{K}$ moves an arc-length element $\mathrm{d}s$ along $\gamma$, it also performs an infinitesimal rotation $\mathrm{d}\boldsymbol{\varphi}$. Expressing this rotation in the basis $\{\mathbf{t}, \mathbf{N}, \mathbf{B}\}$ gives

$$\mathrm{d}\boldsymbol{\varphi} = \alpha_1 \, \mathbf{t} + \alpha_2 \, \mathbf{N} + \alpha_3 \, \mathbf{B} \,, \quad \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R} \,, \tag{3.58}$$

where $\alpha_1$, $\alpha_2$, and $\alpha_3$ are the three aforementioned characteristic quantities that relate the arc-length derivatives $\{\mathbf{t}', \mathbf{N}', \mathbf{B}'\}$ with $\{\mathbf{t}, \mathbf{N}, \mathbf{B}\}$. According to the well-known Euler differentiation rule, the differentials of $\{\mathbf{t}, \mathbf{N}, \mathbf{B}\}$ can be expressed compactly as

$$\mathrm{d}\mathbf{t} = \mathrm{d}\boldsymbol{\delta} \times \mathbf{t} \,, \tag{3.59}$$

$$\mathrm{d}\mathbf{N} = \mathrm{d}\boldsymbol{\delta} \times \mathbf{N} \,, \tag{3.60}$$

$$\mathrm{d}\mathbf{B} = \mathrm{d}\boldsymbol{\delta} \times \mathbf{B} \,. \tag{3.61}$$

Inserting Eqn. 3.58 in Eqn. 3.59, Eqn. 3.60, and Eqn. 3.61, and using the relations $\mathbf{t} = \mathbf{N} \times \mathbf{B}$, $\mathbf{N} = \mathbf{B} \times \mathbf{t}$, and $\mathbf{B} = \mathbf{t} \times \mathbf{N}$ gives

$$\begin{bmatrix} \mathbf{t}' \\ \mathbf{N}' \\ \mathbf{B}' \end{bmatrix} = \begin{bmatrix} 0 & \alpha_3 & -\alpha_2 \\ -\alpha_3 & 0 & \alpha_1 \\ \alpha_2 & -\alpha_1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix} . \tag{3.62}$$

In the literature, $\alpha_1$ is typically called geodesic torsion $\tau_g$, $-\alpha_2$ is called tangential or geodesic curvature $\kappa_g$, and $\alpha_3$ is called normal curvature $\kappa_n$. Using these quantities, the

complete derivative of the Darboux trihedron with respect to the arc length of the curve is given by

$$\begin{bmatrix} \mathbf{x}' \\ \mathbf{t}' \\ \mathbf{N}' \\ \mathbf{B}' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \kappa_n & \kappa_g \\ -\kappa_n & 0 & \tau_g \\ -\kappa_g & -\tau_g & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix} . \tag{3.63}$$

The normal curvature $\kappa_n$ is easily evaluated using Eqn. 3.22. The geodesic torsion $\tau_g$ is compactly expressed as

$$\tau_g = \frac{(\mathrm{E\,M} - \mathrm{F\,L})\,\mathrm{d}u^2 + (\mathrm{E\,N} - \mathrm{G\,L})\,\mathrm{d}u\,\mathrm{d}v + (\mathrm{F\,N} - \mathrm{G\,M})\,\mathrm{d}v^2}{\sqrt{\mathrm{E\,G} - \mathrm{F}^2}\,(\mathrm{E}\,\mathrm{d}u^2 + 2\,\mathrm{F}\,\mathrm{d}u\,\mathrm{d}v + \mathrm{G}\,\mathrm{d}v^2)} . \tag{3.64}$$

Remarkably, both the normal curvature $\kappa_n$ and the geodesic torsion $\tau_g$ depend only on a point and a direction on the surface as shown by Eqn. 3.22 and Eqn. 3.64. Therefore, these quantities are identical for any flock of curves that pass through the same point with the same direction. However, the determination of the geodesic curvature $\kappa_g$ requires more knowledge about the curve considered. It is only zero for geodesics or at inflection points.

For the determination of $\kappa_g$ it is favorable to describe the considered curve in an orthogonal coordinate system, i.e., a coordinate system with $\mathrm{F} = \mathbf{x}_u \cdot \mathbf{x}_v = 0$, as a line with $u$ = constant or $v$ = constant, respectively (Fig. 3.6). In such a coordinate system, the geodesic curvature can be written explicitly in terms of the coefficients E and G of the first fundamental form.

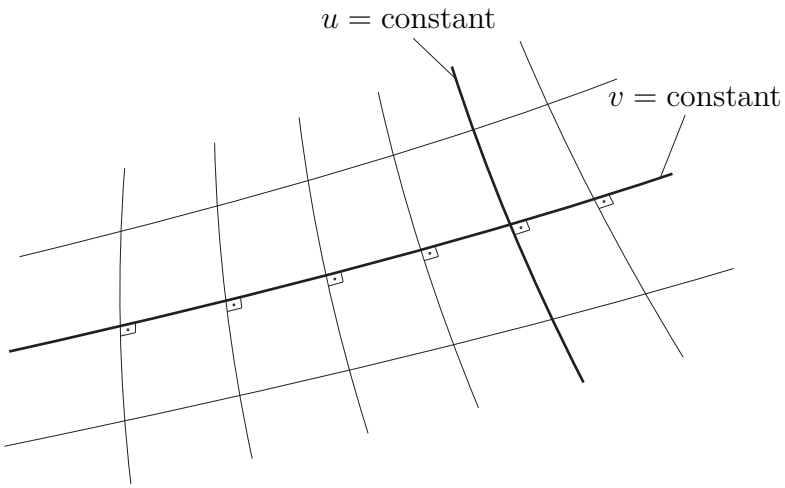
The geodesic curvature along the coordinate curve with $u$ = constant is given by [70]

$$\kappa_g^{u=\text{const.}} = -\frac{\mathrm{E}_v}{2\,\mathrm{E}\sqrt{\mathrm{G}}} , \tag{3.65}$$

while for the curve with $v$ = constant it holds

$$\kappa_g^{v=\text{const.}} = \frac{\mathrm{G}_u}{2\,\mathrm{G}\sqrt{\mathrm{E}}} . \tag{3.66}$$

**Figure 3.6** Orthogonal surface parameterization with $\mathrm{F} = \mathbf{x}_u \cdot \mathbf{x}_v = 0$. The coordinate lines with $u =$ constant intersect the lines with $v =$ constant orthogonally. This simplifies the computation of the geodesic curvature on these lines.




## 3.5 Surface examples

This section contains three examples of surfaces that are well-suited for muscle wrapping applications. It discusses their mathematical properties and illustrates exemplarily the computation of geodesic curves and Darboux trihedra.

### Cylinder

The cylinder is an elementary geometric wrapping surface, typically used for wrapping the triceps muscle over the ellbow. It is often parameterized by

$$\mathbf{x} = \begin{bmatrix} R\cos u \\ R\sin u \\ v \end{bmatrix}, \tag{3.67}$$

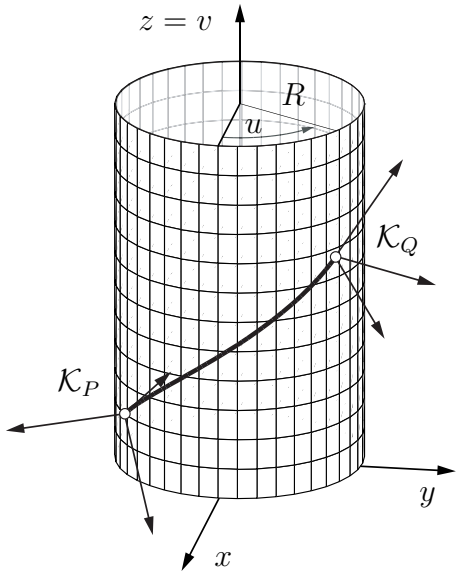
where $R$ is the radius, $u$ is an angle measured from the $x$ axis about the $z$ axis, and $v$ parameterizes the height. Figure 3.7 shows a parametric cylinder with $R = 1$ in the parameter domain $u \in [0, 2\pi[$ and $v \in [0, 3]$. For illustrative purposes, it too shows a geodesic parameterized by

$$\mathbf{q} = [\ -1.0,\ 0.5,\ 0.82,\ 0.57,\ 2.5\ ]^{\mathrm{T}}\ . \tag{3.68}$$

The first two parameters define the curve's start point on the cylinder, the subsequent two define its direction and are chosen such that the geodesic has unit speed, and the last

one is the geodesic's total arc length. Figure 3.7 also shows two Darboux trihedra of the geodesic: $\mathcal{K}_P$ at the start point $P$ and $\mathcal{K}_Q$ it the end point $Q$.

**Figure 3.7** Example of cylinder with a geodesic segment and the two Darboux trihedra at the geodesic's boundary points.




The cylinder has some special properties. For instance, it has constant Gaußian curvature $K = 0$ everywhere because the normal curvature in the direction of the longitudinal $z$ axis is always zero. The coefficients of the first fundamental form are simply $\mathrm{E} = R^2$, $\mathrm{F} = 0$, and $\mathrm{G} = 1$, where the fact that F is equal to zero means that the cylinder's parameterization is orthogonal. Geodesic curves on a cylinders are helixes, which can be computed explicitly.

## Torus

The torus is well-suited to guide musculotendon paths within a certain area. A common parameterization is

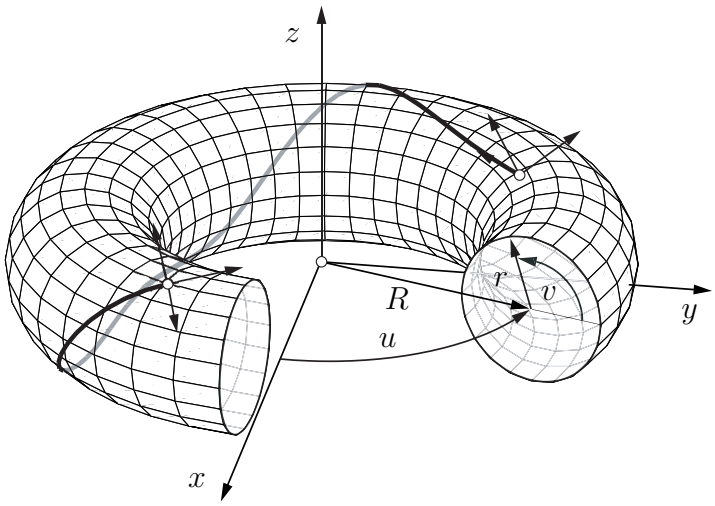
$$\mathbf{x} = \begin{bmatrix} (R + r\,\cos v)\,\cos u \\ (R + r\,\cos v)\,\sin u \\ r\,\sin v \end{bmatrix}, \tag{3.69}$$

where $R$ and $r$ are the two radii of the torus as shown in Fig. 3.8. For the shown torus the two radii were chosen as $R = 3$ and $r = 1$. The exemplarily shown geodesic segment is parameterized by

$$\mathbf{q} = [\ 2.0,\ 2.0,\ 0.37,\ 0.27,\ 14.5\ ]^{\mathrm{T}}\ . \tag{3.70}$$

The parameter domains of $(u, v)$ are $u \in [0, 2\pi[$ and $v \in [0, 2\pi[$.

**Figure 3.8** Example of torus with a geodesic segment and the two Darboux trihedra at the geodesic's boundary points.




The torus is a special closed surface in terms of its Gaußian curvature. It holds $K > 0$ for $3/2\pi < v < \pi/2$, $K = 0$ for $v = \pi/2$ and $v = 3/2\pi$, and $K < 0$ for $\pi/2 < v < 3/2\pi$.

## Fitted surface patches

Fitted surface patches are well-suited to approximate experimentally measured data. When fitting surfaces, the general goal is to find an analytic function that approximates a discrete point cloud to allow for the computation of intrinsic surface properties both at the measured points and between them.

A surface fit depends on the chosen model, i.e., the underlying structure of the analytic function that shall approximate the given data. Typical models are, for instance, non-uniform rational basis splines (NURBS), Bézier splines, or polynomials. Here, a brief presentation of polynomial surfaces is given to convey the main idea of surface fitting.

The general structure of a polynomial surface is

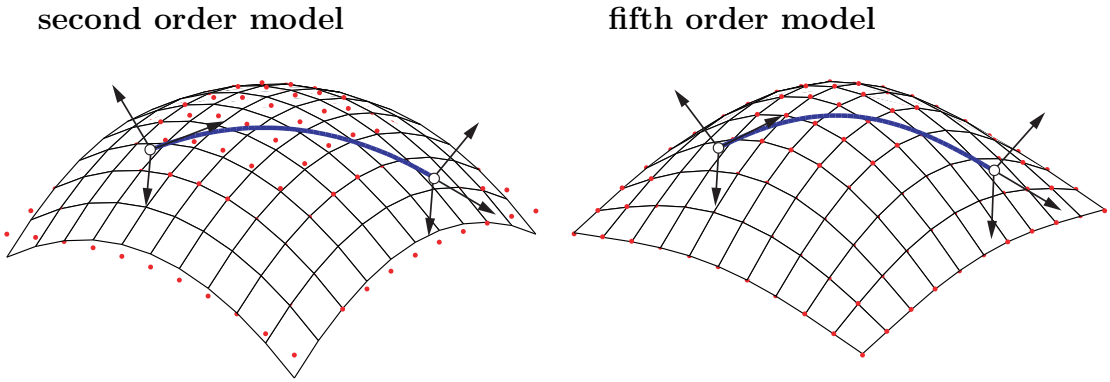
$$f(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_{ij}\, u^i\, v^j\,, \tag{3.71}$$

where $p_{ij}$ are constant coefficients and $n$ and $m$ are the maximum order of $u$ and $v$, respectively. Setting $x = u$, $y = v$, and $z = f$ gives a representation of the surface in cartesian coordinates. Polynomial surfaces with various orders can be generated automatically using Matlab for instance. Automated fitting routines aim to compute the coefficients $p_{ij}$

such that the resulting surface fit has the least sum of squared errors with respect to the given data.

The following example compares two surface fits to the same data with different model complexity: (*i*) a model with order two in $u$ and $v$, and (*ii*) a model with order five in $u$ and $v$ (Fig. 3.9). The target data-set of 121 points was generated by the function $f = \sin u \sin v$, where $u \in [0.1\pi, 0.9\pi]$ and $v \in [0.1\pi, 0.9\pi]$. For illustrative purposes, two geodesic segments with the same initial conditions are also shown on the surfaces.

**Figure 3.9** Example of two polynomial surfaces fitted to discrete data points using a Matlab fitting routine. The left surface is of second order, where the right surface is of fifth order. The sum of squared distances between the data points and the surface is 0.4412 for the second order model and 0.0013 for the fifth order model.




The choice of a sufficient polynomial order depends on the required accuracy of the fit. For the given example data, the second order fit yields a sum of squared errors of 0.4412, while the fifth order fit yields an error of 0.0013. On the downside, the computational costs for evaluating Eqn. 3.71 increase with a higher polynomial order, and a higher polynomial order may yield oscillations when approximating noisy data.

# 4 Solving shortest-path problems using natural geodesic variations

This chapter describes the root-finding approach to solving shortest-path problems. It begins with dismantling the problem into its components and introducing an adequate notation. Then it introduces the path-error function that measures how close a candidate path is from the sought solution. Hereafter, it describes the explicit determination of the path-error Jacobian that is used to iteratively solve for the shortest path. The chapter finishes with explaining how the unknowns of the problem, i.e., the parameters of all geodesics, are updated at each iteration step. To achieve these objectives in a straightforward manner, this chapter builds rigorously upon the differential-geometric foundations of geodesics and moving Darboux trihedra.

## 4.1 Notation

Consider the locally shortest (musculotendon) path between an origin point $O$ and an insertion point $I$, which wraps over an ordered set of $n$ obstacle surfaces $\mathcal{S}^i$ $(i = 1, \ldots, n)$. The minimal-length condition implies that the total path can be subdivided into a concatenation of shortest-path segments: $n$ geodesic segments $\gamma^i$ on the surfaces and $n + 1$ straight-line segments $\sigma^j$ $(j = 1, \ldots, n + 1)$ between the geodesics as well as from $O$ to the start point of $\gamma^1$ and from the end point of $\gamma^n$ to $I$.

The shortest path suffices the condition that each geodesic segment $\gamma^i$ is located on surface $\mathcal{S}^i$ such that it connects collinearly with its two adjacent straight-line segments $\sigma^i$ and $\sigma^{i+1}$. In arbitrary configurations of the geodesic segments on the surfaces, the transitions between geodesic and straight-line segments can be noncollinear (Fig. 4.1). In this case, the shortest-path condition is violated. Thus the collinearity condition allows for formulating the shortest-path problem as one of finding a feasible set of $n$ surface geodesics $\gamma^i$ on the surfaces $\mathcal{S}^i$ such that all geodesics connect collinearly with their adjacent straight-line segments.

Measuring the deviation from collinearity at the transitions requires a metric. This metric, termed path error, is explained in the following section.

## 4.2 Global path error

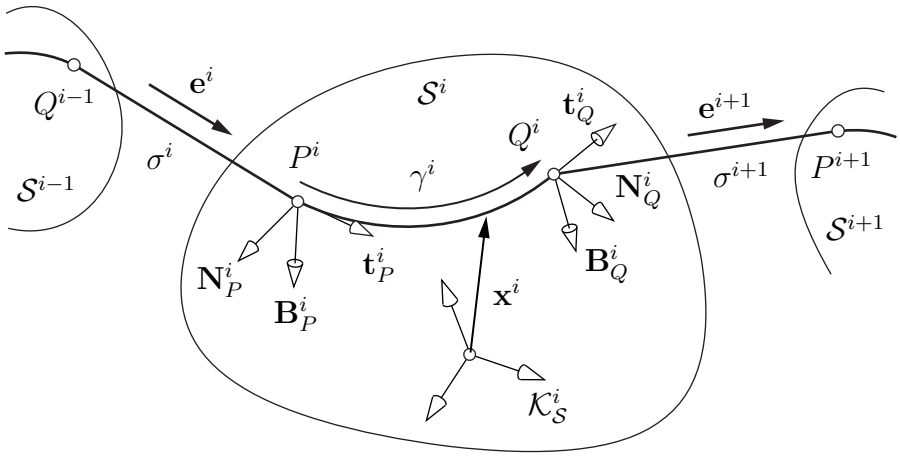
At the path's solution configuration, each geodesic $\gamma^i$ connects collinearly with its two adjacent straight-line segments $\sigma^i$ and $\sigma^{i+1}$. For a local surface geodesic $\gamma^i$, this collinearity condition can be expressed in terms of the two boundary-point trihedra

$$\mathcal{K}_P^i := \left\{\mathbf{x}_P^i,\ \mathbf{t}_P^i,\ \mathbf{N}_P^i,\ \mathbf{B}_P^i\right\}, \tag{4.1}$$

$$\mathcal{K}_Q^i := \left\{\mathbf{x}_Q^i,\ \mathbf{t}_Q^i,\ \mathbf{N}_Q^i,\ \mathbf{B}_Q^i\right\}, \tag{4.2}$$

at the geodesic's start and end points $P^i$ and $Q^i$, and the unit vectors $\mathbf{e}^i$ and $\mathbf{e}^{i+1}$ along $\sigma^i$ and $\sigma^{i+1}$ (Fig. 4.1). For a collinear transition at $P^i$, the projection of $\mathbf{e}^i$ both onto $\mathbf{N}_P^i$ and $\mathbf{B}_P^i$ must vanish. Analogously, the projection of $\mathbf{e}^{i+1}$ onto $\mathbf{N}_Q^i$ and $\mathbf{B}_Q^i$ must vanish for a collinear transition at point $Q^i$. Figure 4.1 illustrates a collinear transition at $P^i$ and a noncollinear transition at $Q^i$.

**Figure 4.1** Example of a collinear transition between a local surface geodesic $\gamma^i$ and the straight-line segment $\sigma^i$ at the geodesic's start point $P^i$, and a noncollinear transition between $\gamma^i$ and $\sigma^{i+1}$ at its end point $Q^i$. The deviation from collinearity of each transition is measured by projecting the unit vectors of adjacent straight-line segments onto the normal and binormal vectors of the geodesic's boundary-point trihedra, yielding a local path-error function $\boldsymbol{\varepsilon}^i$ for each geodesic.




The collinearity conditions allow for stating the local path-error function of geodesic $\gamma^i$ as follows

$$\boldsymbol{\varepsilon}^i(\mathbf{q}^{i-1}, \mathbf{q}^i, \mathbf{q}^{i+1}) := \begin{bmatrix} \mathbf{e}^i & \cdot & \mathbf{N}_P^i \\ \mathbf{e}^i & \cdot & \mathbf{B}_P^i \\ \mathbf{e}^{i+1} & \cdot & \mathbf{N}_Q^i \\ \mathbf{e}^{i+1} & \cdot & \mathbf{B}_Q^i \end{bmatrix}. \tag{4.3}$$

The local path error $\boldsymbol{\varepsilon}^i$ depends on the parameters $\mathbf{q}^i$ of the local surface geodesic, and also on the parameters of neighboring geodesics. That is, it depends on $\mathbf{q}^{i-1}$ if $i > 1$ and on $\mathbf{q}^{i+1}$ if $i < n$. Note that the shortest-path problem is solved at a particular time frame during a simulation, so all wrapping obstacle surfaces $\mathcal{S}^i$ as well as path origin $O$ and insertion $I$ can be considered to be fixed.

Assembling all local path errors $\boldsymbol{\varepsilon}^i$ and all local geodesic parameters $\mathbf{q}^i$ to global vectors $\boldsymbol{\varepsilon}$ and $\mathbf{q}$ yields the global path error $\boldsymbol{\varepsilon}$ as a function of the geodesic parameters $\mathbf{q}$

$$\boldsymbol{\varepsilon}(\mathbf{q}) := \begin{bmatrix} \boldsymbol{\varepsilon}^1 \\ \boldsymbol{\varepsilon}^2 \\ \vdots \\ \boldsymbol{\varepsilon}^n \end{bmatrix} \in \mathbb{R}^{4n\times 1}, \quad \mathbf{q} := \begin{bmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \\ \vdots \\ \mathbf{q}^n \end{bmatrix} \in \mathbb{R}^{5n\times 1}\,. \tag{4.4}$$

The locally shortest path between points $O$ and $I$ is found if the global path error vanishes, which yields the necessary and sufficient condition

$$\boldsymbol{\varepsilon}(\mathbf{q}) = \mathbf{0}\,. \tag{4.5}$$

With the above system of constraints (Eqn. 4.5), the shortest-path problem is cast into a root-finding problem. An efficient solution search requires the path-error Jacobian, i.e., the gradient of the path error with respect to the geodesic parameters. Since the path error is formulated in terms of the boundary-point trihedra $\mathcal{K}_P^i$ and $\mathcal{K}_Q^i$, and in terms of the unit vectors $\mathbf{e}^i$ and $\mathbf{e}^{i+1}$, establishing the path-error Jacobian requires knowledge how these objects vary as functions of the geodesic parameters. This requires the concept of natural geodesic variations, which is described in the next section.

## 4.3 Natural geodesic variations

In Sec. 3.3.3 it is shown that a geodesic segment on a surface is uniquely defined by its start point parameterized by $(u_P, v_P)$, its initial direction parameterized by $(u'_P, v'_P)$, and its total arc length $\ell_\gamma$. Accordingly, the poses of the two boundary-point trihedra $\mathcal{K}_P$ and $\mathcal{K}_Q$ result directly from these five parameters and it holds

$$\mathcal{K}_P = \mathcal{K}_P\,(u_P,\ v_P,\ u'_P,\ v'_P)\ , \tag{4.6}$$

$$\mathcal{K}_Q = \mathcal{K}_Q\,(u_P,\ v_P,\ u'_P,\ v'_P,\ \ell_\gamma)\ . \tag{4.7}$$

The determination of the global path-error Jacobian requires information how $\mathcal{K}_P$ and $\mathcal{K}_Q$ displace on the surface when the geodesic parameters vary. Differentiating $\mathcal{K}_P$ and $\mathcal{K}_Q$ directly with respect to the geodesic parameters is unfavorable because of two major reasons. First, the computed derivatives would depend explicitly on the surface representation. However, different surface representations require a different number of parameters to define a geodesic: five parameters on parametric surfaces and seven parameters on implicit surfaces (Cha. 7). In this setting, additional constraints for dependent parameters must be added, such as the unit-speed constraint (Eqn. 3.54) for the parameters $(u'_P, v'_P)$, to form a regular constraint system. Second, the basis $\{\mathbf{x}_u, \mathbf{x}_v\}_P$ is generally distorted, with $(\mathbf{x}_u)_P$ and $(\mathbf{x}_v)_P$ not being unit vectors.

The approach to avoid the aforementioned complications with derivatives that depend on a surface representation is to introduce a set of four natural variations of the geodesic parameters, which are $(i)$ independent of the surface representation and $(ii)$ independent of each other. These are:

1. the infinitesimal displacement $\mathrm{d}s_P$ of the geodesic's start point $P$ in direction of the tangent $\mathbf{t}_P$ (Fig. 4.2),
2. the infinitesimal displacement $\mathrm{d}\beta_P$ of the geodesic's start point $P$ in direction of the binormal $\mathbf{B}_P$ (Fig. 4.3),
3. the infinitesimal clockwise rotation $\mathrm{d}\theta$ of the geodesic's initial direction about the normal $\mathbf{N}_P$ (Fig. 4.4), and
4. the infinitesimal length increment $\mathrm{d}\ell_\gamma$ of the geodesic's length at point $Q$ for a fixed point $P$ (Fig. 4.5).

The resulting vector $\mathrm{d}\boldsymbol{\xi}$ of natural geodesic variations is then given by

$$\mathrm{d}\boldsymbol{\xi} := \begin{bmatrix} \mathrm{d}s_P & \mathrm{d}\beta_P & \mathrm{d}\theta & \mathrm{d}\ell_\gamma \end{bmatrix}^{\mathrm{T}} . \tag{4.8}$$

These variations are defined in a coordinate system that is induced by the geodesic itself and thereby they do not depend on surface-specific coordinates. Note that for any of the above variations, the remaining three can be kept zero. For instance, a binormal displacement $\mathrm{d}\beta_P$ of the start point $P$ does not cause a tangential displacement $\mathrm{d}s_P$, a rotation $\mathrm{d}\theta$ of the initial direction, or a length increment $\mathrm{d}\ell_\gamma$ of the geodesic. Accordingly, no additional constraints must be considered when working with natural geodesic variations.

In the following, the partial derivatives of $\mathcal{K}_P$ and $\mathcal{K}_Q$ with respect to $\mathrm{d}\boldsymbol{\xi}$ are derived. Note that in this setting, the common notation for differentiation $\partial h/\partial \xi_j$ with $\mathrm{d}\xi_1 = \mathrm{d}s_P$, $\mathrm{d}\xi_2 = \mathrm{d}\beta_P$, $\mathrm{d}\xi_3 = \mathrm{d}\theta$, and $\mathrm{d}\xi_4 = \mathrm{d}\ell_\gamma$ is used even if $h(\xi_j)$ does not exist explicitly. Then

the expression $\partial h/\partial \xi_j$ corresponds to a pseudo derivative that maps variations $\mathrm{d}\xi_j$ to variations $\mathrm{d}h$.

According to each individual variation in $\mathrm{d}\boldsymbol{\xi}$ there exists a variational vector field $\mathbf{J}(s)$ along $\gamma$ which describes how close the corresponding geodesic variation is located with respect to the original geodesic. It can be shown that such variational vector fields fulfill the following Jacobi equation [70]

$$\mathrm{D}^2\mathbf{J}(s) + K(s)\,(\mathbf{t}(s) \times \mathbf{J}(s)) \times \mathbf{t}(s) = \mathbf{0}\,, \tag{4.9}$$

where $K$ is the Gaußian curvature, and D denotes the covariant, or sometimes called absolute, derivative [72, 70, 73] with respect to arc length. The covariant derivative is equal to the usual derivative projected onto the tangent space of $\mathcal{S}$. An important property of Jacobi fields is that they fulfill the following equation [73]

$$\mathbf{J}(s) \cdot \mathbf{t}(s) = \mathbf{J}(0) \cdot \mathbf{t}(0) + (\mathbf{J}'(0) \cdot \mathbf{t}(0))\, s\,, \tag{4.10}$$

which relates their behavior at any point at arc-length $s$ along a geodesic with their behavior at $s = 0$. Both Eqn. 4.9 and Eqn. 4.10 play an important role for the following sections in which the geodesic variations with respect to $\mathrm{d}s_P$, $\mathrm{d}\beta_P$, $\mathrm{d}\theta$, and $\mathrm{d}\ell_\gamma$ are discussed successively.

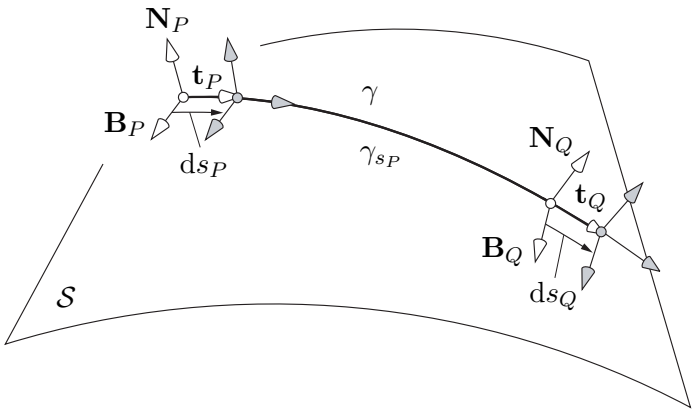
### 4.3.1 Variation in tangential direction

The first natural geodesic variation discussed here is the infinitesimal displacement $\mathrm{d}s_P$ of the geodesic's start point $P$ in direction of the tangent $\mathbf{t}_P$ (Fig. 4.2). For the variation $\mathrm{d}s_P$, the other three variations are considered to be zero. This means that $(i)$ the start point of $\gamma_{s_P}$ lies on $\gamma$, $(ii)$ the tangent of $\gamma_{s_P}$ at $s = 0$ is equal to the tangent of $\gamma$ at $s = \mathrm{d}s_P$, and $(iii)$ the geodesic $\gamma_{s_P}$ has the same length as $\gamma$. Therefore, the variation $\gamma_{s_P}$ partially covers the original geodesic $\gamma$ and the end point $Q$ displaces by $\mathrm{d}s_Q$ in direction of $\mathbf{t}_Q$.

The relation between $\mathrm{d}s_P$ and $\mathrm{d}s_Q$ follows by setting $\mathrm{d}\ell_\gamma = 0$ in Eqn. 3.42

$$\frac{\partial s_Q}{\partial s_P} = 1\,. \tag{4.11}$$

**Figure 4.2** An infinitesimal displacement $\mathrm{d}s_P$ of the geodesic's start point $P$ in direction of the tangent $\mathbf{t}_P$ causes an infinitesimal displacement $\mathrm{d}s_Q = \mathrm{d}s_P$ of the geodesic's end point $Q$ in direction of $\mathbf{t}_Q$.




Thus, according to the Darboux formulas in Eqn. 3.63, the derivatives of $\mathcal{K}_P$ and $\mathcal{K}_Q$ with respect to a tangential displacement of the geodesic's start point are

$$\frac{\partial}{\partial s_P}\begin{bmatrix}\mathbf{x}\\ \mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_{P,Q} = \begin{bmatrix}1 & 0 & 0\\ 0 & \kappa_n & 0\\ -\kappa_n & 0 & \tau_g\\ 0 & -\tau_g & 0\end{bmatrix}^{\gamma}_{P,Q}\begin{bmatrix}\mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_{P,Q}, \tag{4.12}$$

where $\kappa_g^\gamma = 0$ was inserted as $\gamma$ has no geodesic curvature. Here the indices $\{P, Q\}$ indicate that Eqn. 4.12 is valid both for the start point $P$ and the end point $Q$ of $\gamma$ when the corresponding quantities $\kappa_n$ and $\tau_g$ at these points are inserted.

### 4.3.2 Variation in binormal direction

The second variation considered here is the infinitesimal displacement $\mathrm{d}\beta_P$ of the geodesic's start point $P$ in direction of the binormal $\mathbf{B}_P$ (Fig. 4.3). For this variation, it holds $\mathrm{d}s_P = \mathrm{d}\ell_\gamma = 0$ and in particular $\mathrm{d}\theta = 0$. The latter condition demands that the geodesic's initial direction is locally kept constant for the considered variation. This is only fulfilled if $\mathcal{K}_P$ is parallel-transported along a curve with vanishing tangential curvature, i.e., a geodesic $\gamma^\perp$ through $P$ in direction of $\mathbf{B}_P$. At point $P$, the tangent of $\gamma^\perp$ is equal to the

binormal of $\gamma$, the binormal of $\gamma^\perp$ is equal to the negative tangent of $\gamma$, and both surface normal vectors are equal. Thus for $\partial\mathcal{K}_P/\partial\beta_P$ it holds

$$\frac{\partial}{\partial\beta_P}\begin{bmatrix}\mathbf{x}\\ \mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_P = \begin{bmatrix}0 & 0 & 1\\ 0 & \tau_g & 0\\ -\tau_g & 0 & -\kappa_n\\ 0 & \kappa_n & 0\end{bmatrix}_P^\perp \begin{bmatrix}\mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_P , \tag{4.13}$$

where $\kappa_n^\perp$ and $\tau_g^\perp$ are the normal curvature and the geodesic torsion at $P$ in direction of $\mathbf{B}_P$, and $\kappa_g^\perp = 0$ was inserted. Note that two orthogonal curves through one point have the same geodesic torsion but with a different sign. Thus it holds $\tau_{g,P}^\perp = -\tau_{g,P}^\gamma$, which can be shown by computing $\partial\mathbf{t}_P/\partial\beta_P$ explicitly using Young's theorem

$$\frac{\partial\mathbf{t}_P}{\partial\beta_P} = \frac{\partial}{\partial\beta_P}\left(\frac{\partial\mathbf{x}_P}{\partial s_P}\right) = \frac{\partial}{\partial s_P}\left(\frac{\partial\mathbf{x}_P}{\partial\beta_P}\right) = \frac{\partial}{\partial s_P}\left(\mathbf{B}_P\right) = -\tau_{g,P}^\gamma\,\mathbf{N}_P\,. \tag{4.14}$$

Comparing Eqn. 4.12 with Eqn. 4.14 confirms $\tau_{g,P}^\perp = -\tau_{g,P}^\gamma$.

It remains to derive $\partial\mathcal{K}_Q/\partial\beta_P$, which is described by the variational vector field $\mathbf{J}(s) = \mathbf{x}_{\beta_P}(s)$ at $Q$, i.e., at $s = \ell_\gamma$. At $s = 0$ it holds

$$\mathbf{x}_{\beta_P}(0)\cdot\mathbf{t}(0) = 0 \tag{4.15}$$

because $\mathbf{x}_{\beta_P}(0) = \mathbf{B}(0)$. Differentiating the above equation with respect to arc length at $s = 0$ gives

$$\mathbf{x}'_{\beta_P}(0)\cdot\mathbf{t}(0) + \mathbf{x}_{\beta_P}(0)\cdot\mathbf{t}'(0) = \mathbf{x}'_{\beta_P}(0)\cdot\mathbf{t}(0) = 0 \tag{4.16}$$

since $\mathbf{t}'(0)$ is orthogonal to $\mathbf{x}_{\beta_P}(0)$. Substituting $\mathbf{J}(s) = \mathbf{x}_{\beta_P}(s)$ in Eqn. 4.9 and inserting both Eqn. 4.15 and Eqn. 4.16 gives the necessary condition

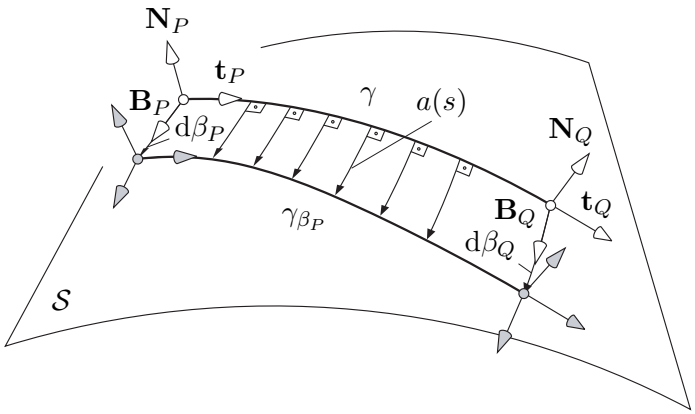
$$\mathbf{x}_{\beta_P}(s)\cdot\mathbf{t}(s) = 0\,,\quad s\in[0,\ell_\gamma]\,, \tag{4.17}$$

for the Jacobi field $\mathbf{x}_{\beta_P}(s)$. The above Eqn. 4.17 shows that $\mathbf{x}_{\beta_P}(s)$ is always orthogonal to the geodesic $\gamma$ (Fig. 4.3), or zero. Accordingly, $\mathbf{x}_{\beta_P}(s)$ is a multiple of some scalar $a(s)$ and the binormal vector $\mathbf{B}(s)$

$$\mathbf{x}_{\beta_P}(s) = a(s)\,\mathbf{B}(s)\,,\quad a(s)\in\mathbb{R}\,,\quad s\in[0,\ell_\gamma]\,, \tag{4.18}$$

as illustrated in Fig. 4.3.

**Figure 4.3** An infinitesimal displacement $\mathrm{d}\beta_P$ of the geodesic's start point $P$ in direction of the binormal $\mathbf{B}_P$ causes an infinitesimal displacement $\mathrm{d}\beta_Q = a\,\mathrm{d}\beta_P$ of the geodesic's end point $Q$ in direction of $\mathbf{B}_Q$.




Inserting $\mathbf{J}(s) = a(s)\,\mathbf{B}(s)$ into the Jacobi Eqn. 4.9 and multiplying it by $\mathbf{B}(s)$ simplifies it to a scalar Jacobi equation

$$a''(s) + K(s)\,a(s) = 0\,, \tag{4.19}$$

with

$$a(s) = \mathbf{x}_{\beta_P}(s) \cdot \mathbf{B}(s)\,. \tag{4.20}$$

Note that $a'(s) = \mathrm{D}a(s)$ according to the rules for covariant differentiation [72].

Solving the scalar Jacobi Eqn. 4.19 requires the initial conditions $a(0)$ and $a'(0)$, where $a(0) = 1$ results directly from inserting $\mathbf{x}_{\beta_P}(0) = \mathbf{B}(0) = \mathbf{B}_P$ into Eqn. 4.20. The second initial condition follows by differentiating Eqn. 4.20

$$a'(s) = \mathrm{D}\mathbf{x}_{\beta_P}(s) \cdot \mathbf{B}(s) + \mathbf{x}_{\beta_P}(s) \cdot \mathrm{D}\mathbf{B}(s)\,. \tag{4.21}$$

At $s = 0$ it holds $\mathbf{x}_{\beta_P}(0) = \mathbf{B}_P$, thus $\mathrm{D}\mathbf{x}_{\beta_P}(0) = \mathbf{0}$. Furthermore it holds $\mathrm{D}\mathbf{B}(s) = \mathbf{0}$ since $\mathbf{B}(s)$ is the parallel transport of $\mathbf{B}_P$ along $\gamma$. It further follows from Eqn. 4.21 that $a'(0) = 0$. Given the initial conditions $a(0) = 1$ and $a'(0) = 0$, the relation between a displacement $\mathrm{d}\beta_P$ at $P$ and the resulting binormal displacement $\mathrm{d}\beta_Q$ is obtained by evaluating Eqn. 4.19 at $s = \ell_\gamma$

$$\frac{\partial \beta_Q}{\partial \beta_P} = a(\ell_\gamma) = a_Q\,. \tag{4.22}$$

The next step is the determination of the derivative $\partial\mathcal{K}_Q/\partial\beta_P$ using the general Darboux formulas in Eqn. 3.63 and the above Eqn. 4.22. Let $\alpha$ be the curve through $Q$ in direction of $\mathbf{B}_Q$ along which $\mathcal{K}_Q$ travels when $P$ is displaced by $\mathrm{d}\beta_P$. At Q, the tangent of $\alpha$ is equal to the binormal of $\gamma$, the binormal of $\alpha$ is equal to the negative tangent of $\gamma$, and both surface normals are equal. Thus it holds for $\partial\mathcal{K}/\partial\beta_Q$ expressed in $\mathcal{K}_Q$

$$\frac{\partial}{\partial\beta_Q}\begin{bmatrix}\mathbf{x}\\ \mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_Q = \begin{bmatrix}0 & 0 & 1\\ 0 & \tau_g & \kappa_g\\ -\tau_g & 0 & -\kappa_n\\ -\kappa_g & \kappa_n & 0\end{bmatrix}^{\alpha}_{Q}\begin{bmatrix}\mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_Q . \tag{4.23}$$

The normal curvature $\kappa_n^\alpha$ follows from Eqn. 3.22, while for the geodesic torsion it holds $\tau_{g,Q}^\alpha = -\tau_{g,Q}^\gamma$ which is known after evaluating Eqn. 4.12. The remaining unknown in Eqn. 4.23 is the geodesic torsion $\kappa_{g,Q}^\alpha$ of $\alpha$ at $Q$. To derive it, consider the neighborhood of the geodesic as being parameterized by $\mathbf{x}(s,\beta_P)$. For this parameterization the first fundamental form is given by

$$\mathrm{E}^{(s,\beta_P)} = \mathbf{x}_s\cdot\mathbf{x}_s = 1 \tag{4.24}$$

$$\mathrm{F}^{(s,\beta_P)} = \mathbf{x}_s\cdot\mathbf{x}_{\beta_P} = 0 \tag{4.25}$$

$$\mathrm{G}^{(s,\beta_P)} = \mathbf{x}_{\beta_P}\cdot\mathbf{x}_{\beta_P} = a^2\,, \tag{4.26}$$

where $\mathrm{F}=0$ means that the parameterization, also known after Gauß as the geodesic binormal coordinate system (Sec. 2.4.1), is orthogonal. As explained in Sec. 3.4, the geodesic curvature of the curve with $s=$ constant, i.e., the curve $\alpha$, is given by [70]

$$\kappa_g^\alpha = \frac{\mathrm{G}_s^{(s,\beta_P)}}{2\,\mathrm{G}^{(s,\beta_P)}\sqrt{\mathrm{E}^{(s,\beta_P)}}} \tag{4.27}$$

Inserting Eqn. 4.24 and Eqn. 4.26 into Eqn. 4.27 yields the compact expression

$$\kappa_g^\alpha = \frac{a_Q'}{a_Q}\,. \tag{4.28}$$

for the sought geodesic curvature. Note that $\kappa_g^\alpha$ is equal to zero at $\ell_\gamma = 0$, which is consistent with the previously made assumption that $\gamma$ is parallel transported along a binormal

geodesic $\gamma^{\perp}$. Inserting Eqn. 4.28 into Eqn. 4.23 and multiplying by $a_Q = \partial\beta_Q/\partial\beta_P$ (Eqn. 4.22) gives the sought derivative $\partial\mathcal{K}_Q/\partial\beta_P$ expressed in $\mathcal{K}_Q$

$$\frac{\partial}{\partial\beta_P}\begin{bmatrix}\mathbf{x}\\ \mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_Q = \begin{bmatrix}0 & 0 & a\\ 0 & -a\,\tau_g^{\gamma} & a'\\ a\,\tau_g^{\gamma} & 0 & -a\,\kappa_n^{\alpha}\\ -a' & a\,\kappa_n^{\alpha} & 0\end{bmatrix}_Q \begin{bmatrix}\mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_Q , \tag{4.29}$$

where $\tau_g^{\alpha} = -\tau_g^{\gamma}$ was inserted.

### 4.3.3 Variation of the initial direction

The third natural geodesic variation considered here is the infinitesimal rotation $\mathrm{d}\theta$ of the geodesic's initial direction, measured in clockwise direction about the surface normal $\mathbf{N}_P$ (Fig. 4.4).

It is straightforward to show that at $P$

$$\frac{\partial}{\partial\theta}\begin{bmatrix}\mathbf{x}\\ \mathbf{t}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}_P = \begin{bmatrix}\mathbf{0}\\ \mathbf{B}\\ \mathbf{0}\\ -\mathbf{t}\end{bmatrix}_P . \tag{4.30}$$

It is of further interest how the variation $\mathrm{d}\theta$ affects the trihedron $\mathcal{K}_Q$, which is given by the Jacobi field $\mathbf{x}_\theta$ at $Q$. The derivation of the derivative $\partial\mathcal{K}_Q/\partial\theta$ is similar to the derivation of $\partial\mathcal{K}_Q/\partial\beta_P$ described in Sec. 4.3.2, thus this section is kept brief. Again, the behavior of the sought Jacobi field $\mathbf{x}_\theta$ is first regarded at $s = 0$ and then at $s = \ell_\gamma$. Equation 4.30 shows that at $s = 0$

$$\mathbf{x}_\theta(0) = \mathbf{0}\,. \tag{4.31}$$

Since $\mathbf{x}'(0)$ is equal to the geodesic's tangent $\mathbf{t}_P$ at $P$, it holds by Eqn. 4.30 that

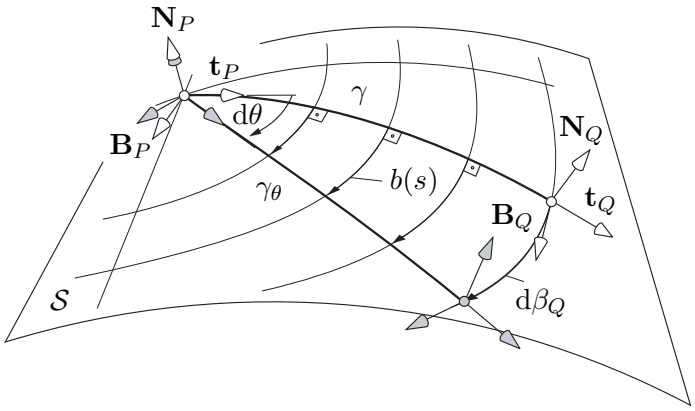
$$\mathbf{x}'_\theta(0) = \mathbf{B}_P\,. \tag{4.32}$$

Substituting $\mathbf{J}(s) = \mathbf{x}_\theta(s)$ in Eqn. 4.10 and inserting Eqn. 4.31 and Eqn. 4.32 gives the necessary condition

$$\mathbf{x}_\theta(s)\cdot\mathbf{t}(s) = 0\,, \quad s \in [0, \ell_\gamma]\ , \tag{4.33}$$

for the variation $\mathbf{x}_\theta$. Equation 4.33 is also known as the lemma of Gauß [70], which ensures that $\mathbf{x}_\theta$ is always orthogonal to the geodesic's tangent, or zero at points conjugate to $P$. Accordingly, all points on the geodesic keep a constant geodesic distance to point $P$ when the initial direction is rotated. The curves with constant geodesic distance to $P$ are commonly called geodesic circles.

**Figure 4.4** An infinitesimal clockwise rotation $\mathrm{d}\theta$ of the geodesic's initial direction about the surface normal $\mathbf{N}_P$ causes a binormal displacement $\mathrm{d}\beta_Q$ of the end-point trihedron $\mathcal{K}_Q$. The relation $r_Q = \partial\beta_Q/\partial\theta$ results as the solution of the scalar Jacobi Eqn. 4.35, which is solved along the geodesic.




By the lemma of Gauß and analog to Eqn. 4.18, $\mathbf{x}_\theta$ can be written as

$$\mathbf{x}_\theta(s) = r(s)\,\mathbf{B}(s)\,, \quad r(s) \in \mathbb{R}\,, \quad s \in [0, \ell_\gamma]\,, \tag{4.34}$$

where $r$ is some scalar that remains to be determined. Inserting $\mathbf{J}(s) = r(s)\,\mathbf{B}(s)$ into the Jacobi Eqn. 4.9 and multiplying it by $\mathbf{B}(s)$ simplifies it to a scalar Jacobi equation that is analog to Eqn. 4.19

$$r''(s) + K(s)\,r(s) = 0\,. \tag{4.35}$$

Solving this scalar Jacobi Eqn. 4.35 requires the initial conditions $r(0)$ and $r'(0)$. With $\mathrm{D}\mathbf{B}(s) = \mathbf{0}$ it holds

$$r(s) = \mathbf{x}_\theta(s) \cdot \mathbf{B}(s)\,, \tag{4.36}$$

$$r'(s) = \mathrm{D}\mathbf{x}_\theta(s) \cdot \mathbf{B}(s)\,, \tag{4.37}$$

which together with Eqn. 4.31 and Eqn. 4.32 give $r(0) = 0$ and $r'(0) = 1$. Finally, the relationship between $\mathrm{d}\theta$ and $\mathrm{d}\beta_Q$ along the geodesic circle with $\ell_\gamma = \text{constant}$ is obtained by evaluating Eqn. 4.35 at $s = \ell_\gamma$

$$\frac{\partial \beta_Q}{\partial \theta} = r(\ell_\gamma) = r_Q \,. \tag{4.38}$$

Let $\rho(\theta)$ denote the geodesic circle along which $\mathcal{K}_Q$ moves when the geodesic's initial direction is rotated. The parameterization $\mathbf{x}(s, \theta)$ is orthogonal, thus the geodesic curvature of $\rho(\theta)$ is given by (Sec. 3.4)

$$\kappa_g^\rho = \frac{\mathrm{G}_s^{(s,\theta)}}{2\,\mathrm{G}^{(s,\theta)}\sqrt{E^{(s,\theta)}}} \,, \tag{4.39}$$

where

$$\mathrm{E}^{(s,\theta)} = \mathbf{x}_s \cdot \mathbf{x}_s = 1 \,, \tag{4.40}$$

$$\mathrm{F}^{(s,\theta)} = \mathbf{x}_s \cdot \mathbf{x}_\theta = 0 \,, \tag{4.41}$$

$$\mathrm{G}^{(s,\theta)} = \mathbf{x}_\theta \cdot \mathbf{x}_\theta = r^2 \,. \tag{4.42}$$

Inserting Eqn. 4.40 and Eqn. 4.42 into Eqn. 4.39 gives

$$\kappa_g^\rho = \frac{r'_Q}{r_Q} \tag{4.43}$$

and thus for the derivative $\partial \mathcal{K}_Q / \partial \theta$ expressed in $\mathcal{K}_Q$ it holds

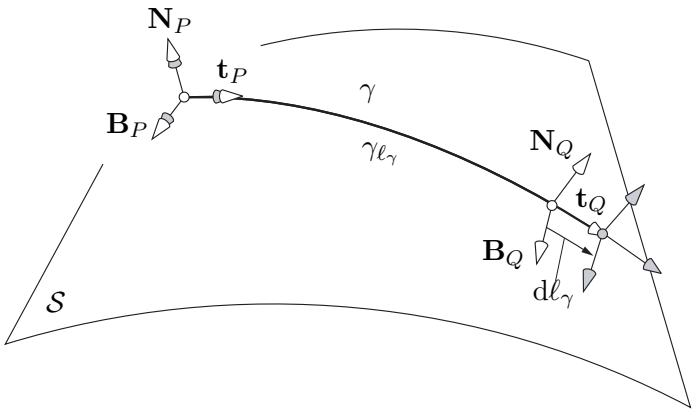
$$\frac{\partial}{\partial \theta} \begin{bmatrix} \mathbf{x} \\ \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix}_Q = \begin{bmatrix} 0 & 0 & r \\ 0 & -r\,\tau_g^\gamma & r' \\ r\,\tau_g^\gamma & 0 & -r\,\kappa_n^\rho \\ -r' & r\,\kappa_n^\rho & 0 \end{bmatrix}_Q \begin{bmatrix} \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix}_Q \,. \tag{4.44}$$

Note that for $\ell_\gamma = 0$ it holds $P = Q$. At this point $r(0) = 0$ and $r'(0) = 1$. Inserting these conditions into the above Eqn. 4.44 confirms Eqn. 4.30.

### 4.3.4 Variation of length

The fourth and last natural geodesic variation discussed in this section is the infinitesimal length increment $\mathrm{d}\ell_\gamma$ of the geodesic at point $Q$ for a fixed point $P$ and a fixed initial direction at $P$ (Fig. 4.5).

**Figure 4.5** An infinitesimal length increment $\mathrm{d}\ell_\gamma$ of the geodesic with a fixed start point $P$ causes a tangential displacement $\mathrm{d}s_Q = \mathrm{d}\ell_\gamma$ of the end point $Q$. This variation does not affect the trihedron $\mathcal{K}_P$.




As shown in Fig. 4.5 the trihedron $\mathcal{K}_P$ is not affected by this variation. Thus it simply holds

$$\frac{\partial}{\partial \ell_\gamma} \begin{bmatrix} \mathbf{x} \\ \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix}_P = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{4.45}$$

At the other end of the geodesic, the point $Q$ displaces in direction of $\mathbf{t}_Q$. The derivative $\partial \mathcal{K}_Q / \partial \ell_\gamma$ results directly from the Darboux formulas (Eqn. 3.63)

$$\frac{\partial}{\partial \ell_\gamma} \begin{bmatrix} \mathbf{x} \\ \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix}_Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \kappa_n & 0 \\ -\kappa_n & 0 & \tau_g \\ 0 & -\tau_g & 0 \end{bmatrix}_Q^\gamma \begin{bmatrix} \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix}_Q \tag{4.46}$$

and is equal to the derivative $\partial \mathcal{K}_Q / \partial s_P$ given in Eqn. 4.12.

The derivatives of $\mathcal{K}_P$ and $\mathcal{K}_Q$ with respect to the four natural geodesic variations provide the necessary differential-geometric foundations to establish the global path-error Jacobian, which is discussed next.

## 4.4 Global path-error Jacobian

The global path-error Jacobian $\mathbf{J} := \partial \boldsymbol{\varepsilon} / \partial \boldsymbol{\xi}$ maps natural geodesic variations to path-error variations. Thus its inverse contains information about how to correct the geodesic pa-

rameters such that the path error vanishes. This section describes the overall structure of the global path-error Jacobian and its assembly from boundary-point trihedra derivatives explained in the previous section.

A general block row $i$ in $\mathbf{J}$ contains elements $\partial\boldsymbol{\varepsilon}^i/\partial\boldsymbol{\xi}^i$ that represent local path-error Jacobians. It also contains elements $\partial\boldsymbol{\varepsilon}^i/\partial\boldsymbol{\xi}^{i-1}$ for $i > 1$ and $\partial\boldsymbol{\varepsilon}^i/\partial\boldsymbol{\xi}^{i+1}$ for $i < n$ because of the dependency of local path errors from the parameters of neighboring geodesics. Accordingly, the Jacobian $\mathbf{J}$ has block band structure with five scalar lower and three scalar upper diagonals

$$\mathbf{J} = \begin{bmatrix} \frac{\partial\boldsymbol{\varepsilon}^1}{\partial\boldsymbol{\xi}^1} & \frac{\partial\boldsymbol{\varepsilon}^1}{\partial\boldsymbol{\xi}^2} & & & & & \\ \frac{\partial\boldsymbol{\varepsilon}^2}{\partial\boldsymbol{\xi}^1} & \frac{\partial\boldsymbol{\varepsilon}^2}{\partial\boldsymbol{\xi}^2} & \frac{\partial\boldsymbol{\varepsilon}^2}{\partial\boldsymbol{\xi}^3} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \frac{\partial\boldsymbol{\varepsilon}^i}{\partial\boldsymbol{\xi}^{i-1}} & \frac{\partial\boldsymbol{\varepsilon}^i}{\partial\boldsymbol{\xi}^i} & \frac{\partial\boldsymbol{\varepsilon}^i}{\partial\boldsymbol{\xi}^{i+1}} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \frac{\partial\boldsymbol{\varepsilon}^{n-1}}{\partial\boldsymbol{\xi}^{n-2}} & \frac{\partial\boldsymbol{\varepsilon}^{n-1}}{\partial\boldsymbol{\xi}^{n-1}} & \frac{\partial\boldsymbol{\varepsilon}^{n-1}}{\partial\boldsymbol{\xi}^n} \\ & & & & & \frac{\partial\boldsymbol{\varepsilon}^n}{\partial\boldsymbol{\xi}^{n-1}} & \frac{\partial\boldsymbol{\varepsilon}^n}{\partial\boldsymbol{\xi}^n} \end{bmatrix}. \tag{4.47}$$

The $j$-th column ($j = 1, \ldots, 4$) of the $i$-th local path-error Jacobian is given by

$$\frac{\partial\boldsymbol{\varepsilon}^i}{\partial\boldsymbol{\xi}^i_j} = \begin{bmatrix} \frac{\partial\mathbf{N}^i_P}{\partial\xi^i_j}\cdot\mathbf{e}^i & + & \mathbf{N}^i_P\cdot\frac{\partial\mathbf{e}^i}{\partial\xi^i_j} \\ \frac{\partial\mathbf{B}^i_P}{\partial\xi^i_j}\cdot\mathbf{e}^i & + & \mathbf{B}^i_P\cdot\frac{\partial\mathbf{e}^i}{\partial\xi^i_j} \\ \frac{\partial\mathbf{N}^i_Q}{\partial\xi^i_j}\cdot\mathbf{e}^{i+1} & + & \mathbf{N}^i_Q\cdot\frac{\partial\mathbf{e}^{i+1}}{\partial\xi^i_j} \\ \frac{\partial\mathbf{B}^i_Q}{\partial\xi^i_j}\cdot\mathbf{e}^{i+1} & + & \mathbf{B}^i_Q\cdot\frac{\partial\mathbf{e}^{i+1}}{\partial\xi^i_j} \end{bmatrix}, \tag{4.48}$$

where $\mathrm{d}\xi_1^i = \mathrm{d}s_P^i$, $\mathrm{d}\xi_2^i = \mathrm{d}\beta_P^i$, $\mathrm{d}\xi_3^i = \mathrm{d}\theta^i$, and $\mathrm{d}\xi_4^i = \mathrm{d}\ell_\gamma^i$. The $j$-th columns ($j = 1, \ldots, 4$) of the $i$-th coupling matrices are given by

$$\frac{\partial \boldsymbol{\varepsilon}^i}{\partial \boldsymbol{\xi}_j^{i-1}} = \begin{bmatrix} \mathbf{N}_P^i \cdot \dfrac{\partial \mathbf{e}^i}{\partial \xi_j^{i-1}} & \mathbf{B}_P^i \cdot \dfrac{\partial \mathbf{e}^i}{\partial \xi_j^{i-1}} & 0 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{4.49}$$

$$\frac{\partial \boldsymbol{\varepsilon}^i}{\partial \boldsymbol{\xi}_j^{i+1}} = \begin{bmatrix} 0 & 0 & \mathbf{N}_Q^i \cdot \dfrac{\partial \mathbf{e}^{i+1}}{\partial \xi_j^{i+1}} & \mathbf{B}_Q^i \cdot \dfrac{\partial \mathbf{e}^{i+1}}{\partial \xi_j^{i+1}} \end{bmatrix}^{\mathrm{T}}. \tag{4.50}$$

The above Eqn. 4.48, Eqn. 4.49, and Eqn. 4.50 contain partial derivatives of the vectors $\mathbf{N}_P^i$, $\mathbf{B}_P^i$, $\mathbf{N}_Q^i$, and $\mathbf{B}_Q^i$ with respect to the natural geodesic variations $\mathrm{d}\boldsymbol{\xi}^i$. These derivatives are given explicitly in Sec. 4.3. They also contain partial derivatives of the unit vectors along the straight-line segments with respect to the natural geodesic variations, which can also be determined explicitly using the formulas derived in Sec. 4.3.

The unit vectors $\mathbf{e}^i$ and $\mathbf{e}^{i+1}$ connecting to geodesic $\gamma^i$ are given by

$$\mathbf{e}^i = \frac{\mathbf{x}_P^i - \mathbf{x}_Q^{i-1}}{\| \mathbf{x}_P^i - \mathbf{x}_Q^{i-1} \|}, \tag{4.51}$$

$$\mathbf{e}^{i+1} = \frac{\mathbf{x}_P^{i+1} - \mathbf{x}_Q^i}{\| \mathbf{x}_P^{i+1} - \mathbf{x}_Q^i \|}. \tag{4.52}$$

Note that $\mathbf{e}^1$ is defined using the position of the origin point $O$ as there exists no $\mathbf{x}_Q^0$, while $\mathbf{e}^{n+1}$ is defined using the position of the insertion point $I$ as there exists no $\mathbf{x}_P^{n+1}$. The nonzero derivatives of the unit vectors in Eqn. 4.51 and Eqn. 4.52 with respect to $\mathrm{d}\boldsymbol{\xi}^{i-1}$, $\mathrm{d}\boldsymbol{\xi}^i$, and $\mathrm{d}\boldsymbol{\xi}^{i+1}$ are given by

$$\frac{\partial \mathbf{e}^i}{\partial \xi_j^i} = \frac{\dfrac{\partial \mathbf{x}_P^i}{\partial \xi_j^i} - \mathbf{e}^i \left( \mathbf{e}^i \cdot \dfrac{\partial \mathbf{x}_P^i}{\partial \xi_j^i} \right)}{\ell^i}, \tag{4.53}$$

$$\frac{\partial \mathbf{e}^i}{\partial \xi_j^{i-1}} = \frac{-\dfrac{\partial \mathbf{x}_Q^{i-1}}{\partial \xi_j^{i-1}} + \mathbf{e}^i \left( \mathbf{e}^i \cdot \dfrac{\partial \mathbf{x}_Q^{i-1}}{\partial \xi_j^{i-1}} \right)}{\ell^i}, \tag{4.54}$$

$$\frac{\partial \mathbf{e}^{i+1}}{\partial \xi_j^i} = \frac{-\dfrac{\partial \mathbf{x}_Q^i}{\partial \xi_j^i} + \mathbf{e}^{i+1} \left( \mathbf{e}^{i+1} \cdot \dfrac{\partial \mathbf{x}_Q^i}{\partial \xi_j^i} \right)}{\ell^{i+1}}, \tag{4.55}$$

$$\frac{\partial \mathbf{e}^{i+1}}{\partial \xi_j^{i+1}} = \frac{\dfrac{\partial \mathbf{x}_P^{i+1}}{\partial \xi_j^{i+1}} - \mathbf{e}^{i+1} \left( \mathbf{e}^{i+1} \cdot \dfrac{\partial \mathbf{x}_P^{i+1}}{\partial \xi_j^{i+1}} \right)}{\ell^{i+1}}, \tag{4.56}$$

where $\ell^i$ is the length of the $i$-th straight-line segment. The above Eqn. 4.53, Eqn. 4.54, Eqn. 4.55, and Eqn. 4.56 can be evaluated explicitly using the derivatives of the Darboux

trihedra given in Sec. 4.3. The next section provides details about how the path-error Jacobian is used to iteratively correct the geodesic parameters.

## 4.5 Updating geodesic parameters according to natural geodesic corrections

The global path-error Jacobian $\mathbf{J}$ allows for iteratively zeroing the global path error $\boldsymbol{\varepsilon}$ by correcting the geodesic parameters according to the current path error. Since the Jacobian is formulated in terms of natural geodesic variations, a set of finite natural geodesic corrections is computed at each iteration step. This section describes how these corrections are mapped back to a set of surface-dependent geodesic parameters and describes the overall solution flow.

At a given iteration step $k$, the iterative Newton solver computes a global vector

$$\Delta\boldsymbol{\xi}_{(k)} := -\mathbf{J}_{(k)}^{-1}\,\boldsymbol{\varepsilon}_{(k)} \tag{4.57}$$

of finite natural path corrections to correct the parameters of all local geodesics. For a local geodesic $\gamma^i$ the finite corrections are given by

$$\Delta\boldsymbol{\xi}^i_{(k)} := \begin{bmatrix} \Delta s^i_P & \Delta\beta^i_P & \Delta\theta^i & \Delta\ell^i_\gamma \end{bmatrix}^{\mathrm{T}}_{(k)}, \tag{4.58}$$

These corrections are used to compute a new set of geodesic parameters $\mathbf{q}_{(k+1)}$, which converge to a vanishing path error $\boldsymbol{\varepsilon}$. In the following, the corrections of a single geodesic's start-point position, direction, and length is described. The index $i$ is omitted for the sake of notational simlicity.

### 4.5.1 Start-point position correction

There are two natural geodesic corrections $(\Delta s_P, \Delta\beta_P)_{(k)}$ that correspond to corrections of the geodesic's start point position, which is parameterized by the two surface coordinates $(u_P, v_P)_{(k)}$. The mapping from these two natural corrections to a new pair of surface coordinates $(u_P, v_P)_{(k+1)}$ requires the general coordinate transformation $\partial(u,v)/\partial(s,\beta)$. The sought coordinate transformation can be derived by expressing the differential $\mathrm{d}\mathbf{x}$ both in the surface basis $\{\mathbf{x}_u, \mathbf{x}_v\}$ and in the natural basis $\{\mathbf{t}, \mathbf{B}\}$

$$\mathrm{d}\mathbf{x} = \begin{bmatrix} \mathbf{x}_u & \mathbf{x}_v \end{bmatrix} \begin{bmatrix} \mathrm{d}u \\ \mathrm{d}v \end{bmatrix} = \begin{bmatrix} \mathbf{t} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathrm{d}s \\ \mathrm{d}\beta \end{bmatrix}. \tag{4.59}$$

Pairwise multiplication of Eqn. 4.59 with $\mathbf{x}_u$ and $\mathbf{x}_v$ and solving for $(\mathrm{d}u, \mathrm{d}v)$ yields the sought coordinate transformation $\partial(u,v)/\partial(s,\beta)$. It is given by

$$\begin{bmatrix} \mathrm{d}u \\ \mathrm{d}v \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{E} & \mathrm{F} \\ \mathrm{F} & \mathrm{G} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{t}\cdot\mathbf{x}_u & \mathbf{B}\cdot\mathbf{x}_u \\ \mathbf{t}\cdot\mathbf{x}_v & \mathbf{B}\cdot\mathbf{x}_v \end{bmatrix}}_{\partial(u,v)/\partial(s,\beta)} \begin{bmatrix} \mathrm{d}s \\ \mathrm{d}\beta \end{bmatrix} . \tag{4.60}$$

Thus for the new coordinates $(u_P, v_P)_{(k+1)}$ of the start point $P_{(k+1)}$ it holds

$$\begin{bmatrix} u_P \\ v_P \end{bmatrix}_{(k+1)} = \begin{bmatrix} u_P \\ v_P \end{bmatrix}_{(k)} + \frac{\partial(u_P, v_P)}{\partial(s_P, \beta_P)}_{(k)} \begin{bmatrix} \Delta s_P \\ \Delta\beta_P \end{bmatrix}_{(k)} . \tag{4.61}$$

### 4.5.2 Initial direction correction

The goal is to map the natural correction $\Delta\theta_{(k)}$ of the geodesic's initial direction to a new set of geodesic parameters $(u'_P, v'_P)_{(k+1)}$ that describe the geodesic's initial direction at the new point $(u_P, v_P)_{(k+1)}$. This is done in two steps. The first step consists in rotating the tangent vector $\mathbf{t}_{P,(k)}$ at point $P_{(k)}$ by the computed angular difference $\Delta\theta_{(k)}$ (Fig. 4.6). For the rotated vector $\hat{\mathbf{t}}_{P,(k)}$ at $P_{(k)}$ it holds

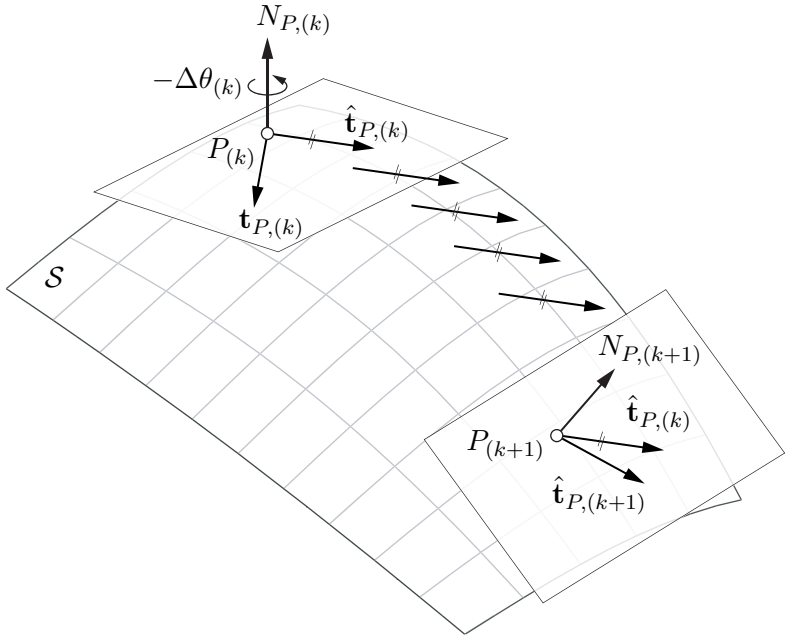
$$\hat{\mathbf{t}}_{P,(k)} = \mathbf{t}_{P,(k)} \cos\Delta\theta_{(k)} + \mathbf{B}_{P,(k)} \sin\Delta\theta_{(k)} . \tag{4.62}$$

In the second step, $\hat{\mathbf{t}}_{P,(k)}$ is parallel transported to the new point $P_{(k+1)}$ with a constant orientation in space according to the Levi-Civita parallelism [72], as illustrated in Fig. 4.6.

At the new point $P_{(k+1)}$, $\hat{\mathbf{t}}_{P,(k)}$ is not tangent to the surface anymore. It has two tangential components with the magnitudes $\mu$, $\nu$, and one normal component with the magnitude $\psi$. Thus $\hat{\mathbf{t}}_{P,(k)}$ can be written as

$$\hat{\mathbf{t}}_{P,(k)} = \begin{bmatrix} \mathbf{x}_u & \mathbf{x}_v & \mathbf{N} \end{bmatrix}_{P,(k+1)} \begin{bmatrix} \mu \\ \nu \\ \psi \end{bmatrix} . \tag{4.63}$$

**Figure 4.6** Updating the the geodesic's initial direction in three steps: (*i*) rotation of $\mathbf{t}_{P,(k)}$ with the angle $\Delta\theta_{(k)}$ about the surface normal $\mathbf{N}_{P,(k)}$ at point $P_{(k)}$; (*ii*) parallel transport of the rotated vector $\hat{\mathbf{t}}_{P,(k)}$ to the new point $P_{(k+1)}$; (*iii*) normalization of $\hat{\mathbf{t}}_{P,(k)}$ such that it is a unit tangent vector at $P_{(k+1)}$.




The next step is to obtain the tangential components $\mu$ and $\nu$ from Eqn. 4.63. Pairwise scalar multiplication of Eqn. 4.63 with $(\mathbf{x}_u)_{P,(k+1)}$ and $(\mathbf{x}_v)_{P,(k+1)}$ cancels out the normal component $\psi$ and yields

$$\begin{bmatrix} \hat{\mathbf{t}}_{P,k} \cdot (\mathbf{x}_u)_{P,(k+1)} \\ \hat{\mathbf{t}}_{P,k} \cdot (\mathbf{x}_v)_{P,(k+1)} \end{bmatrix} = \begin{bmatrix} \mathrm{E} & \mathrm{F} \\ \mathrm{F} & \mathrm{G} \end{bmatrix}_{P,(k+1)} \begin{bmatrix} \mu \\ \nu \end{bmatrix} . \tag{4.64}$$

Thus for the tangential components $\mu$ and $\nu$ of $\hat{\mathbf{t}}_{P,(k)}$ at $P_{(k+1)}$ it holds

$$\begin{bmatrix} \mu \\ \nu \end{bmatrix} = \begin{bmatrix} \mathrm{E} & \mathrm{F} \\ \mathrm{F} & \mathrm{G} \end{bmatrix}^{-1}_{P,(k+1)} \begin{bmatrix} \hat{\mathbf{t}}_{P,(k)} \cdot (\mathbf{x}_u)_{P,(k+1)} \\ \hat{\mathbf{t}}_{P,(k)} \cdot (\mathbf{x}_v)_{P,(k+1)} \end{bmatrix} . \tag{4.65}$$

The third step consists in normalizing $\mu$ and $\nu$ such that $\gamma_{(k+1)}$ has unit speed. The new components $u'_{P,(k+1)}$ and $v'_{P,(k+1)}$ that yield a unit-speed geodesic starting at $P_{(k+1)}$ are given by

$$\begin{bmatrix} u'_P \\ v'_P \end{bmatrix}_{(k+1)} = \left( \frac{1}{\mathrm{E}\,\mu^2 + 2\,\mathrm{F}\,\mu\,\nu + \mathrm{G}\,\nu^2} \right)_{P,(k+1)} \begin{bmatrix} \mu \\ \nu \end{bmatrix} . \tag{4.66}$$

### 4.5.3 Length correction

The computation of the new length $\ell_{\gamma,(k+1)}$ is straightforward

$$\ell_{\gamma,(k+1)} = \ell_{\gamma,(k)} + \Delta\ell_{\gamma,(k)} \,. \tag{4.67}$$

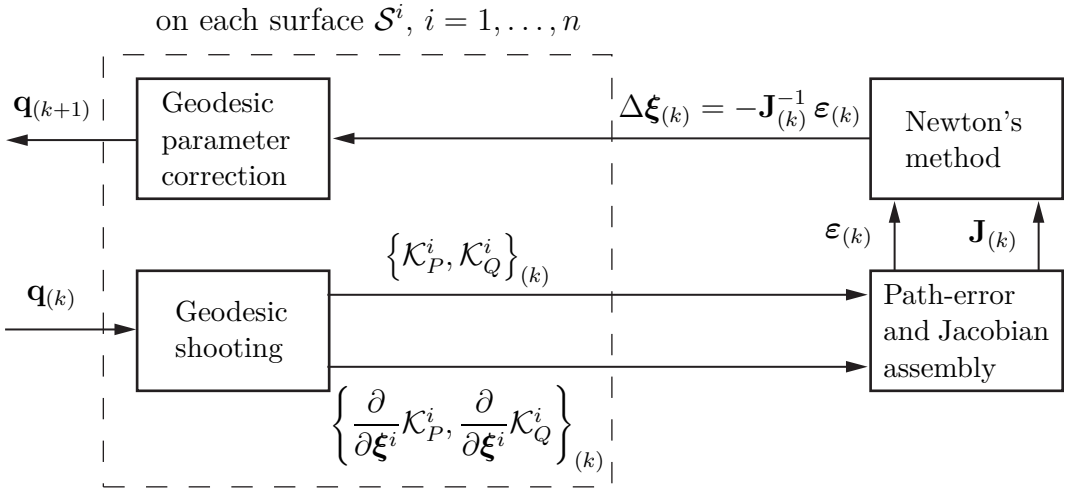
## 4.6 Solution algorithm

The previous sections address the path corrections from a local differential-geometric perspective. This section describes the overall iteration procedure as a whole. It highlights the four necessary steps within a single path iteration step that are required to solve shortest-path problems using natural geodesic variations. For these steps, the simulation time is constant, i.e., the steps are executed while all wrapping obstacle surfaces as well as origin and insertion points are fixed. Figure 4.7 illustrates the four steps. These are:

- **Step 1: Geodesic shooting.** Given a current set $\mathbf{q}_{(k)}$ of geodesic parameters, compute all geodesic segments on the wrapping surfaces and solve the two associated scalar Jacobi equations. Use the geodesics and the solutions of the Jacobi equations to compute the two trihedra $\mathcal{K}_P^i$ and $\mathcal{K}_Q^i$ at each geodesic's boundary points as well as their derivatives $\partial\mathcal{K}_P^i/\partial\boldsymbol{\xi}^i$ and $\partial\mathcal{K}_Q^i/\partial\boldsymbol{\xi}^i$ with respect to the four natural geodesic variations $\mathrm{d}\boldsymbol{\xi}^i$.
- **Step 2: Path-error and Jacobian assembly.** Compute the unit vectors $\mathbf{e}^j$ along the resulting straight-line segments $\sigma^j$ between neighboring geodesics and to path origin $O$ and insertion $I$, as well as their derivatives with respect to the natural geodesic variations. Then evaluate the global path error $\boldsymbol{\varepsilon}$ and assemble the global path-error Jacobian $\mathbf{J}$.
- **Step 3: Newton's method.** Compute a vector $\Delta\boldsymbol{\xi}_{(k)}$ of finite natural geodesic corrections using the global path-error Jacobian and the current path error.
- **Step 4: Geodesic parameter correction.** Compute a new set $\mathbf{q}_{(k+1)}$ of geodesic parameters using the finite natural geodesic corrections $\Delta\boldsymbol{\xi}_{(k)}$.

The shown iteration procedure (Fig. 4.7) must be executed at each simulation time frame until the path error is sufficiently close to zero to compute the locally shortest path. For each iteration, the simulation time remains unchanged so that all obstacle surfaces as well as the origin and insertion points are fixed. The simulation can only continue after the iterations have converged and a feasible shortest path for the current time step is found.

**Figure 4.7** General solution flow of a single iteration step $k$. For a given set of geodesic parameters $\mathbf{q}_{(k)}$, a geodesic engine computes all local surface geodesics, the boundary-point trihedra and their derivatives with respect to the natural geodesic variations. From the spatial positions of the boundary point trihedra and their derivatives, the global path error and the Jacobian are assembled. According to Newton's method, a linear equation system is solved to compute the natural corrections $\Delta\boldsymbol{\xi}_{(k)}$. The natural corrections $\Delta\boldsymbol{\xi}_{(k)}$ are mapped back to a new set of surface-specific geodesic parameters $\mathbf{q}_{(k+1)}$ for the subsequent iteration step $k+1$.




Good convergence during simulation is achieved by using the solution parameters of the previous time step as the initial guess for the current time step. Because of the temporal coherence of subsequent simulation time steps, this feedback ensures convergence in a few iterations. Note that when using Newton's method, quadratic convergence is achieved near the solution, which allows for zeroing the path error with machine precision in a few iteration steps.

From the block diagram (Fig. 4.7) it is visible that the method's computational costs depend linearly on the number of wrapping surfaces. This is because the computational costs of each subtask grow linearly with the number of surfaces. In particular, this is possible because of the banded structure of $\mathbf{J}$, which permits a very efficient, i.e., with order $(n)$, computation of the finite path corrections for $n$ wrapping surfaces when using a linear equation solver that can work with banded matrices.

# 5 Rate of length change of shortest paths

The previous chapters consider a single generic simulation time step and explain the iterative computation of the shortest path at this particular time step. This section addresses the explicit computation of the path's rate of length change as it wraps over a series of moving obstacle surfaces.

A shortest path that wraps around $n$ obstacles can be regarded as a concatenation of $n+1$ straight-line segments $\sigma^j$ of length $\ell^j$ and $n$ geodesic segments $\gamma^i$ of length $\ell^i_\gamma$. The total length $L$ of a shortest path is thus given by the sum of the individual segment lengths

$$L := \sum_{j=1}^{n+1} \ell^j + \sum_{i=1}^{n} \ell^i_\gamma \,. \tag{5.1}$$

Using the dot notation to indicate a derivative with respect to time $t$, the rate of length change $\dot{L}$ of the total path is given by

$$\dot{L} := \sum_{j=1}^{n+1} \dot{\ell}^j + \sum_{i=1}^{n} \dot{\ell}^i_\gamma \,, \quad \dot{(\cdot)} := \frac{\mathrm{d}}{\mathrm{d}t}(\cdot) \,, \tag{5.2}$$

where $\dot{\ell}^j$ is the rate of length change of straight-line segment $\sigma^j$ and $\dot{\ell}^i_\gamma$ is the rate of length change of geodesic $\gamma^i$.

The rate of length change of the $j$-th straight-line segment between two geodesics results directly from projecting its absolute boundary-point velocities $\mathbf{v}_Q^{i-1}$ and $\mathbf{v}_P^i$ on the unit vector $\mathbf{e}^j$

$$\dot{\ell}^j = \mathbf{e}^j \cdot \left[\mathbf{v}_P^j - \mathbf{v}_Q^{j-1}\right] . \tag{5.3}$$

For the first and last straight-line segment it holds

$$\dot{\ell}^1 = \mathbf{e}^1 \cdot \left[\mathbf{v}_P^1 - \mathbf{v}_O\right] , \tag{5.4}$$

$$\dot{\ell}^{n+1} = \mathbf{e}^{n+1} \cdot \left[\mathbf{v}_I - \mathbf{v}_Q^n\right] , \tag{5.5}$$

where $\mathbf{v}_O$ and $\mathbf{v}_I$ are the absolute velocities of the origin point $O$ and the insertion point $I$, respectively.

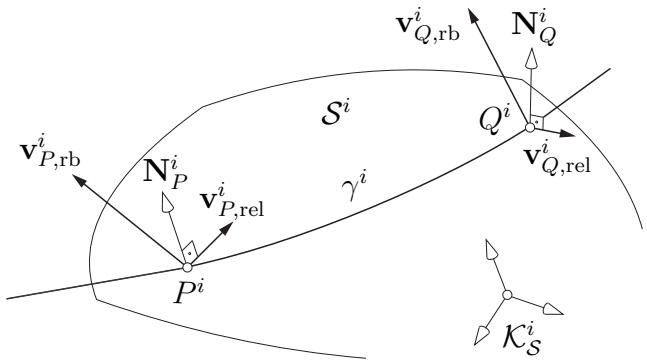
The rate of length change $\dot{\ell}^i_\gamma$ of geodesic $\gamma^i$ results from Eqn. 3.41 as

$$\dot{\ell}^i_\gamma = \mathbf{t}^i_Q \cdot \mathbf{v}^i_{Q,\mathrm{rel}} - \mathbf{t}^i_P \cdot \mathbf{v}^i_{P,\mathrm{rel}}\,, \tag{5.6}$$

where $\mathbf{v}^i_{P,\mathrm{rel}}$ and $\mathbf{v}^i_{Q,\mathrm{rel}}$ are the relative velocities of the boundary points $P^i$ and $Q^i$ with respect to the surface $\mathcal{S}^i$ as illustrated in Fig. 5.1. Since the considered path is locally length minimizing, i.e., the path error is exactly zero, it holds $\mathbf{t}^i_P = \mathbf{e}^i$ and $\mathbf{t}^i_Q = \mathbf{e}^{i+1}$. Thus the above Eqn. 5.6 can be written in terms of the unit vectors $\mathbf{e}^i$ and $\mathbf{e}^{i+1}$ as

$$\dot{\ell}^i_\gamma = \mathbf{e}^{i+1} \cdot \mathbf{v}^i_{Q,\mathrm{rel}} - \mathbf{e}^i \cdot \mathbf{v}^i_{P,\mathrm{rel}}\,. \tag{5.7}$$

**Figure 5.1** Decomposition of absolute boundary-point velocities $\mathbf{v}^i_P$ and $\mathbf{v}^i_Q$ into two components: absolute rigid-body velocities $\mathbf{v}^i_{P,\mathrm{rb}}$, $\mathbf{v}^i_{Q,\mathrm{rb}}$ of the surface at $P^i$ and $Q^i$ and relative boundary-point velocites $\mathbf{v}^i_{P,\mathrm{rel}}$, $\mathbf{v}^i_{Q,\mathrm{rel}}$ of the geodesic's boundary points with respect to the surface.




The relation between the absolute boundary-point velocities in Eqn. 5.3 and the relative velocities in Eqn. 5.7 is given by

$$\mathbf{v}^i_P = \mathbf{v}^i_{P,\mathrm{rb}} + \mathbf{v}^i_{P,\mathrm{rel}}\,, \tag{5.8}$$

$$\mathbf{v}^i_Q = \mathbf{v}^i_{Q,\mathrm{rb}} + \mathbf{v}^i_{Q,\mathrm{rel}}\,, \tag{5.9}$$

where $\mathbf{v}^i_{P,\mathrm{rb}}$ and $\mathbf{v}^i_{Q,\mathrm{rb}}$ are the absolute rigid-body velocities of the surface beneath the geodesic segment $\gamma^i$ at $P^i$ and $Q^i$. Inserting Eqn. 5.8 and Eqn. 5.9 into Eqn. 5.3, Eqn. 5.4, and Eqn. 5.5 yields

$$\dot{\ell}^j = \mathbf{e}^j \cdot \left[\mathbf{v}^j_{P,\mathrm{rb}} + \mathbf{v}^j_{P,\mathrm{rel}} - \mathbf{v}^{j-1}_{Q,\mathrm{rb}} - \mathbf{v}^{j-1}_{Q,\mathrm{rel}}\right], \quad 1 < j < n+1, \tag{5.10}$$

$$\dot{\ell}^1 = \mathbf{e}^1 \cdot \left[\mathbf{v}^1_{P,\mathrm{rb}} + \mathbf{v}^1_{P,\mathrm{rel}} - \mathbf{v}_O\right], \quad j = 1, \tag{5.11}$$

$$\dot{\ell}^{n+1} = \mathbf{e}^{n+1} \cdot \left[\mathbf{v}_I - \mathbf{v}^n_{Q,\mathrm{rb}} - \mathbf{v}^n_{Q,\mathrm{rel}}\right], \quad j = n+1. \tag{5.12}$$

Inserting Eqn. 5.10, Eqn. 5.11, Eqn. 5.12, and Eqn. 5.7 into Eqn. 5.2 shows that all terms that contain relative velocities $\mathbf{v}^i_{P,\mathrm{rel}}$ and $\mathbf{v}^i_{Q,\mathrm{rel}}$ cancel out. Hence Eqn. 5.2 simplifies to

$$\dot{L} = -\mathbf{e}^1 \cdot \mathbf{v}_O + \left(\sum_{i=1}^{n} \mathbf{e}^i \cdot \mathbf{v}^i_{P,\mathrm{rb}} - \mathbf{v}^i_{Q,\mathrm{rb}} \cdot \mathbf{e}^{i+1}\right) + \mathbf{e}^{n+1} \cdot \mathbf{v}_I. \tag{5.13}$$

Accordingly, the rate of length change $\dot{L}$ depends only on the rigid-body velocities $\mathbf{v}^i_{P,\mathrm{rb}}$ and $\mathbf{v}^i_{Q,\mathrm{rb}}$ of the surface beneath the geodesic boundary points $P$ and $Q$, and also on the absolute velocities of the origin and insertion points $O$ and $I$. Equation 5.13 is independent of the formulation used to compute the path itself and computationally inexpensive.

# 6 Methods for simulating path lift-off and touchdown

In the previous chapters is assumed that all wrapping obstacles are in contact with the path. This chapter describes the mathematical foundations for simulating path lift-off and touch-down. A continuous scalar witness function is introduced that allows for identifying when an obstacle makes or breaks contact with the path. These transitions are characterized by the special situation in which the considered surface touches the path at a single point. For the sake of simplicity, the derivations are described by considering the interaction between a single generic surface and a single generic path segment.

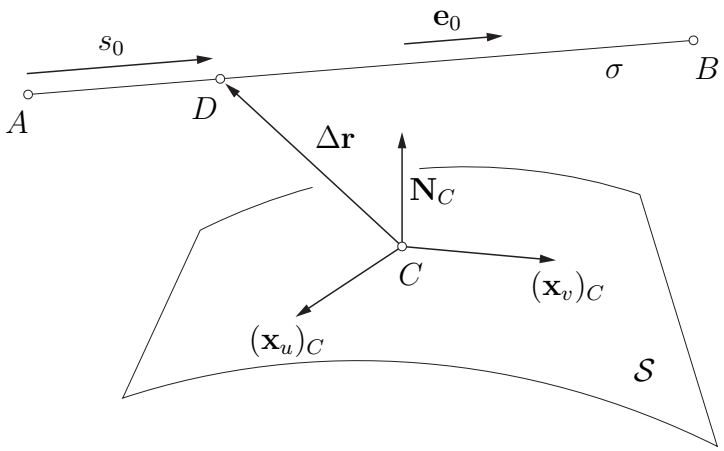
## 6.1 Signed-distance function

The simulation of path lift-off and touchdown requires a continuous scalar function, the so called witness function, which identifies whether an obstacle touches the path or not. In this work, the signed distance $d$ between a generic path segment and a generic wrapping surface in the neighborhood of a given point on the surface is used. It is defined as follows:

- $d > 0$ : the path is not in contact with the surface,
- $d = 0$ : the path touches the surface at a single point,
- $d < 0$ : the path wraps over the surface.

In the following, an iterative method for the determination of $d$ is derived. To this end, consider a generic obstacle surface $\mathcal{S}$ and a straight-line segment $\sigma$ between two points $A$ and $B$ with unit vector $\mathbf{e}_0$ (Fig. 6.1). Point $A$ can be either the end point of the previous adjacent geodesic segment, or the origin point of the path. Point $B$ can be either the start point of the subsequent adjacent geodesic segment or the insertion point of the path. Note that for any surface there exist two points $A$ and $B$, regardless of whether it touches the path or not.

**Figure 6.1** Computation of the signed distance $d$ between a surface $\mathcal{S}$ and a straight-line segment $\sigma$ between two points $A$ and $B$ with unit vector $\mathbf{e}_0$. The goal is to compute a point $C$ on $\mathcal{S}$ and a point $D$ on $\sigma$ such that the vector $\Delta\mathbf{r}$ from $C$ to $D$ is both orthogonal to the straight-line segment and to the surface. Then it holds $d = \boldsymbol{\Delta}\mathbf{r} \cdot \mathbf{N}_C$, where $\mathbf{N}_C$ is the outward surface normal at $C$.



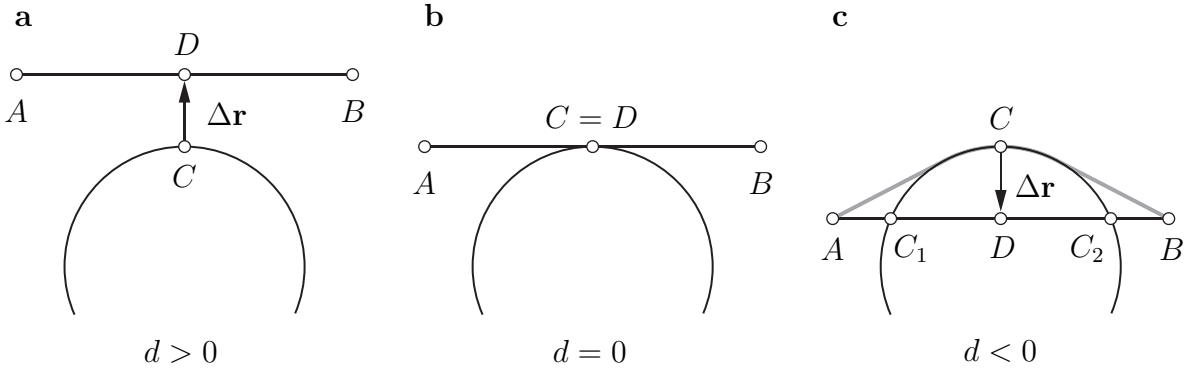

The goal is to compute a point $C$ on the surface $\mathcal{S}$ and a point $D$ on the straight-line segment $\sigma$ such that the vector $\Delta\mathbf{r} = \mathbf{r}_D - \mathbf{r}_C$ from $C$ to $D$ is both orthogonal to $\sigma$ and to $\mathcal{S}$. At this target configuration, the signed distance is given by

$$d := \Delta\mathbf{r} \cdot \mathbf{N}_C \,, \tag{6.1}$$

where $\mathbf{N}_C$ is the outward surface normal at $C$. There exist three different cases for $d$ as mentioned previously. The first possible case occurs when $d > 0$. In that case the vector $\Delta\mathbf{r}$ points away from the surface, i.e., in the direction of the normal at $C$ (Fig. 6.2a). In the second case, where $d = 0$, it holds $C = D$ and the vector $\Delta\mathbf{r}$ is equal to zero (Fig. 6.2b). The third case occurs when $d < 0$. Then the straight-line segment between $A$ and $B$ intersects the surface at two points $C_1$ and $C_2$, and the shortest path between $A$ and $B$ consists of two straight-line segments and one geodesic segment (Fig. 6.2c). In that case $\Delta\mathbf{r}$ points inside the surface, i.e., in the opposite direction of the surface normal at $C$. Note that only in the former two cases the quantity $d$ corresponds to a nonnegative absolute distance in the classical sense. In the third case, the absolute distance between the straight-line segment $\sigma$ and the surface is zero because the path intersects the surface, yet the signed distance is negative.

To compute the signed distance $d$, the points $C$ and $D$ must be found such that the vector $\Delta\mathbf{r}$ is orthogonal to the surface and to the straight-line segment. The position of the point $C$ on the surface is given by its two surface coordinates $(u_C, v_C)$, while the

**Figure 6.2** Three cases for the signed distance $d$. **(a)** $d > 0$: the straight-line segment does not touch the surface and $\Delta\mathbf{r}$ points away from the surface; **(b)** $d = 0$: the straight-line segment touches the surface at a single point and $\Delta\mathbf{r} = \mathbf{0}$; **(c)** $d < 0$: the straight-line segment touches the surface at two points $C_1$ and $C_2$ and $\Delta\mathbf{r}$ points inside the surface.




position of the point $D$ is given by its distance $s_0$ from point $A$ along $\sigma$ (Fig. 6.1). The vector of unknowns is thus

$$\mathbf{w} := \begin{bmatrix} s_0 \\ u_C \\ v_C \end{bmatrix} . \tag{6.2}$$

At the sought configuration, $\mathbf{w}$ fulfills the three orthogonality constraints

$$\mathbf{g}(\mathbf{w}) := \begin{bmatrix} \mathbf{e}_0 & \cdot & \Delta\mathbf{r} \\ (\mathbf{x}_u)_C & \cdot & \Delta\mathbf{r} \\ (\mathbf{x}_v)_C & \cdot & \Delta\mathbf{r} \end{bmatrix} = \mathbf{0} . \tag{6.3}$$

The first row in Eqn. 6.3 enforces that $\Delta\mathbf{r}$ is orthogonal to the straight-line segment, while the second and third rows enforce that $\Delta\mathbf{r}$ is orthogonal to the surface. Eqn. 6.3 must be solved numerically as there is no closed-form solution available in general.

In the following, an iterative solution scheme using the explicit constraint Jacobian is proposed. The Jacobian of $\mathbf{g}$ with respect to the three unknowns in $\mathbf{w}$ is

$$\frac{\partial \mathbf{g}}{\partial \mathbf{w}} = \begin{bmatrix} -\mathbf{e}_0 \cdot \mathbf{x}_u & -\mathbf{e}_0 \cdot \mathbf{x}_v & 1 \\ -\mathrm{E} + \Delta\mathbf{r} \cdot \mathbf{x}_{uu} & -\mathrm{F} + \Delta\mathbf{r} \cdot \mathbf{x}_{uv} & \mathbf{e}_0 \cdot \mathbf{x}_u \\ -\mathrm{F} + \Delta\mathbf{r} \cdot \mathbf{x}_{uv} & -\mathrm{G} + \Delta\mathbf{r} \cdot \mathbf{x}_{vv} & \mathbf{e}_0 \cdot \mathbf{x}_v \end{bmatrix} , \tag{6.4}$$

where E, F and G are the coefficients of the first fundamental form of $\mathcal{S}$. Due to the temporal coherence of the shortest-path problem, it suffices to apply Newton's method to

iteratively zero the constrains in Eqn. 6.3. The iteration procedure for a single iteration step $k$ is given as

$$\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} - \left(\frac{\partial \mathbf{g}}{\partial \mathbf{w}}\right)^{-1}_{(k)} \mathbf{g}_{(k)} \,. \tag{6.5}$$

Repeating the above iteration procedure until $\| \mathbf{g} \|$ is below a specified tolerance yields the vectors $\Delta\mathbf{r}$ and $\mathbf{N}_C$ at the solution configuration. With these vectors, Eqn. 6.1 can be evaluated to determine the signed distance $d$.

### 6.1.1 Example

This example illustrates the iterative computation of the signed distance between a straight-line segment and a parametric paraboloid. Let the positions of points $A$ and $B$ with respect to the paraboloid be given by $\mathbf{r}_A = \begin{bmatrix} -10 & 10 & 0 \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{r}_B = \begin{bmatrix} 10 & 5 & 3 \end{bmatrix}^{\mathrm{T}}$ and let the paraboloid be defined by the equation

$$\mathbf{x}(u, v) = \begin{bmatrix} u \\ v \\ u^2/14 + v^2/6 \end{bmatrix} . \tag{6.6}$$

For this example the following initial values were chosen

$$\mathbf{w}_{(1)} = \begin{bmatrix} s_0 \\ u_C \\ v_C \end{bmatrix}_{(1)} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} , \tag{6.7}$$

where $s_0$ measures the position of point $D$ on the straight-line segment with respect to point $A$ (Fig. 6.3), and $u_C$ and $v_C$ parameterize the position of point $C$ on the paraboloid. Figure 6.3 shows the example setup as well as the positions of points $C$ and $D$ during the iterations.

Table 6.1 displays the values of $s_0$, $u_C$, and $v_C$ for five iterations. It also shows the corresponding error, i.e., the norm $\| \mathbf{g} \|$ of the constraint Eqn. 6.3, and the value of the signed distance $d$. It is clearly visible from the table that only a few iterations are necessary to set $\| \mathbf{g} \|$ sufficiently close to zero as it converges quadratically near the solution.

**Figure 6.3** Iterative computation of the signed distance between a straight-line segment and a paraboloid. Shown are the initial guess $C_{(1)}$ and $D_{(1)}$, the first iteration $C_{(2)}$ and $D_{(2)}$, and the solution $C_{(5)}$ and $D_{(5)}$. Note that the iterations $C_{(3)}$ and $C_{(4)}$ as well as $D_{(3)}$ and $D_{(4)}$ are not visible in the figure due to their proximity to $C_{(5)}$ and $D_{(5)}$, respectively.

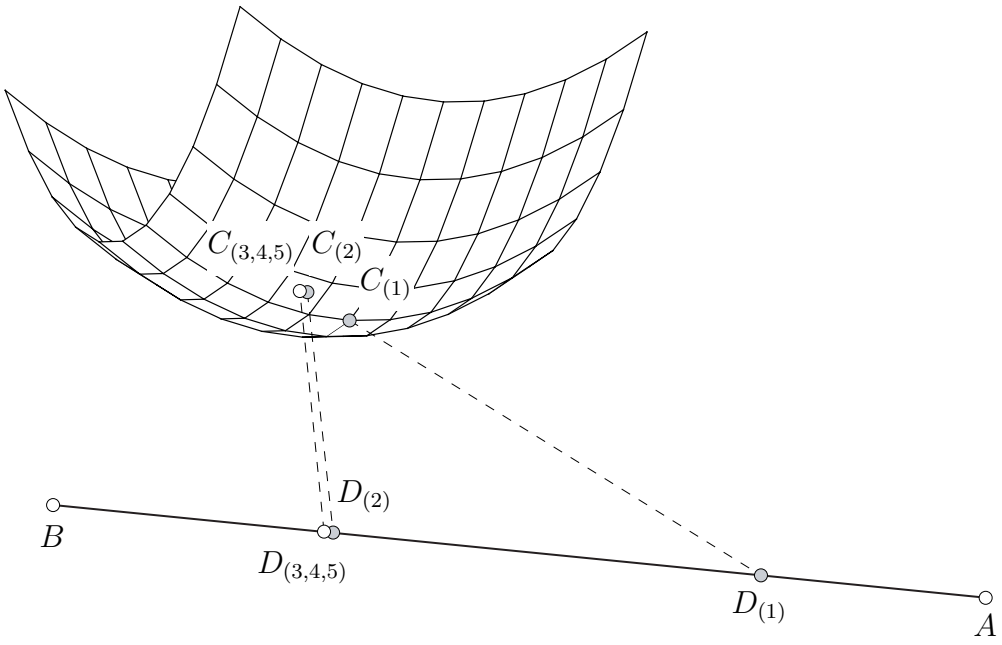



**Table 6.1** Numerical results for the signed distance computation example. The position of point $C$ on the surface and the position of point $D$ on the straight-line segment are computed iteratively such that the distance between $C$ and $D$ is minimal. The history of the error norm $\| \mathbf{g} \|$ shows that quadratic convergence is achieved near the solution, allowing for a high-precision solution in a few iterations.

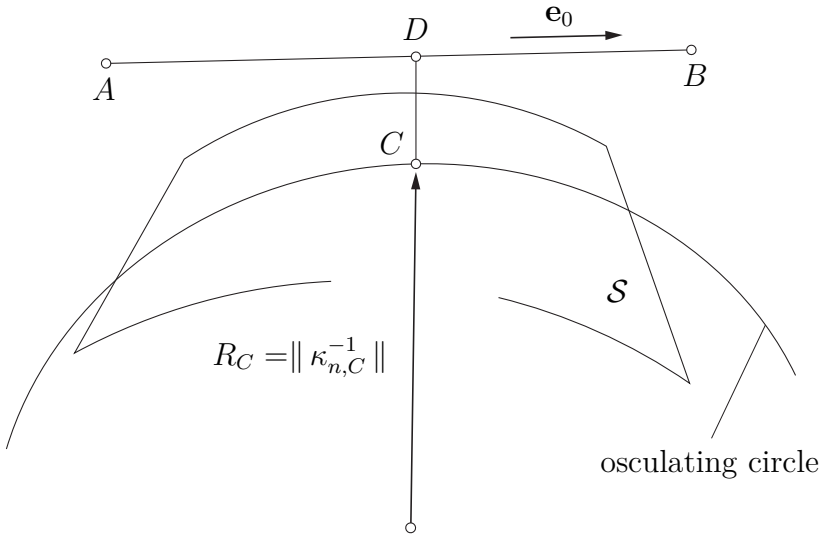
| iteration | parameters | | | error norm | sign. dist. |
|---|---|---|---|---|---|
| $k$ | $s_0$ | $u_C$ | $v_C$ | $\| \mathbf{g} \|$ | $d$ |
| 1 | 5.0000 | 1.0000 | 2.0000 | 11.347973868163585 | 10.4813 |
| 2 | 14.5843 | 2.3944 | 2.6228 | 0.089960279340279 | 6.1210 |
| 3 | 14.7876 | 2.5635 | 2.5943 | 0.002075502462570 | 6.1193 |
| 4 | 14.7875 | 2.5631 | 2.5938 | 0.000000170549412 | 6.1193 |
| 5 | 14.7875 | 2.5630 | 2.5938 | 0.000000000000001 | 6.1193 |

## 6.2 Tolerances for the signed distance

For numerical simulations, it is advantageous to trigger contact events at a certain tolerance for $d$ rather than exactly at $d = 0$. That is, to assume that the path is in contact with the obstacle when $d < d_{\text{tol}}$ and not in contact when $d > d_{\text{tol}}$, where $d_{\text{tol}}$ is a negative value. This tolerance is necessary to avoid rattling effects with short geodesic segments.

The first step to ascertaining a reasonable value for $d_{\text{tol}}$ is to normalize $d_{\text{tol}}$ with respect to the curvature of the surface. The goal of this normalization is to make the contact detection independent of the surface dimension. For instance, a large surface may require a large value for $d_{\text{tol}}$ while a smaller surface may require a small value for $d_{\text{tol}}$. The normalization of $d_{\text{tol}}$ is performed by utilizing the normal curvature $\kappa_{n,C}$ of the surface at the closest point $C$ in the direction of the unit vector $\mathbf{e}_0$ along the straight-line segment $\sigma$ (Fig. 6.4).

**Figure 6.4** Estimation of the surface dimension by considering the normal curvature $\| \kappa_{n,C} \|$ at point $C$ in the direction of the straight-line unit vector $\mathbf{e}_0$. The inverse $\| \kappa_{n,C}^{-1} \|$ corresponds to the radius of the osculating circle at $C$ in direction of $\mathbf{e}_0$, which allows for relating the signed distance tolerance $d_{\text{tol}}$ to the unit circle.




The tolerance $d_{\text{tol}}$ can be related to the unit circle by using the norm $\| \kappa_{n,C} \|$ because its reciprocal equals the radius $R_C$ of the osculating circle at $C$ in direction of $\mathbf{e}_0$ (Fig. 6.4). This yields the following necessary conditions for making and breaking contact

$$d \| \kappa_{n,C} \| \leq d_{\text{tol}} \quad \ldots \quad \text{contact}\,, \tag{6.8}$$

$$d \| \kappa_{n,C} \| > d_{\text{tol}} \quad \ldots \quad \text{no contact}\,, \tag{6.9}$$

where $d$ is the signed distance between the surface and the considered straight-line segment. In practical applications, $d_{\text{tol}}$ has to be chosen as close to zero as possible to avoid discontinuities in path length and rate of length change. These discontinuities can occur because of the instantaneous switching from a straight-line segment to a concatenation of two straight-line segments and one geodesic segment after touchdown.

## 6.3 Transmission of forces

This section describes how tensional muscle force $F_{\mathrm{T}}$ is transmitted to wrapping surfaces that touch the path. Recall here that there is no friction between the muscle path and the wrapping surface and therefore $F_{\mathrm{T}}$ is constant along the muscle. Also, the muscle does not transmit tangential forces to the surface by this assumption.

Consider a single generic surface $\mathcal{S}$ with a single generic geodesic segment $\gamma$ with start point $P$ and end point $Q$ (Fig. 6.5). The tensional muscle force at any point of the path is $F_{\mathrm{T}}$. This force is transmitted to the surface according to the geodesic's curvature $\kappa$ in the direction of the geodesic's normal $\mathbf{n}$. For an infinitesimal normal force element $\mathrm{d}\mathbf{F}$ it holds

$$\mathrm{d}\mathbf{F} = F_{\mathrm{T}}\,\mathbf{n}\,\kappa\,\mathrm{d}s\,, \tag{6.10}$$

and thus for an infinitesimal moment element it holds

$$\mathrm{d}\mathbf{M} = \mathbf{x} \times \mathrm{d}\mathbf{F}\,. \tag{6.11}$$

Integrating the above Eqn. 6.10 and Eqn. 6.11 with respect to arc length gives the resulting force $\mathbf{F}$ and the resulting moment $\mathbf{M}$ which are applied to the surface by the shortest path

$$\mathbf{F} = F_{\mathrm{T}} \int_0^{\ell_\gamma} \mathbf{n}\,\kappa\,\mathrm{d}s\,, \tag{6.12}$$

$$\mathbf{M} = F_{\mathrm{T}} \int_0^{\ell_\gamma} \mathbf{x} \times \mathbf{n}\,\kappa\,\mathrm{d}s\,, \tag{6.13}$$

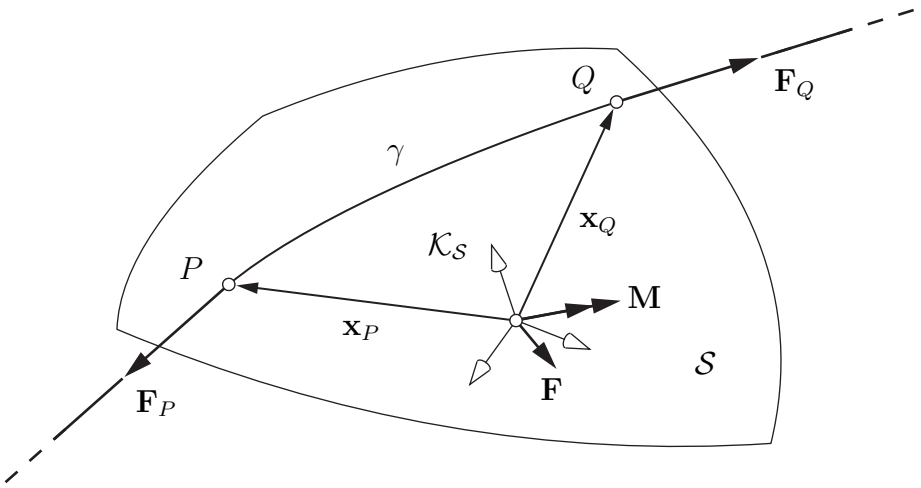
where $\ell_\gamma$ is the geodesic's total arc length. Evaluating $\mathbf{F}$ and $\mathbf{M}$ using the above Eqn. 6.12 and Eqn. 6.13 requires numerical integration and is therefore computationally expensive. However, since the shortest path is stationary at any instant of time, $\mathbf{F}$ and $\mathbf{M}$ can also be determined directly by using a free-body diagram of the geodesic segment (Fig. 6.5).

The forces $\mathbf{F}_P$ and $\mathbf{F}_Q$ at the geodesic's boundary points are given by

$$\mathbf{F}_P = -F_{\mathrm{T}}\,\mathbf{t}_P\,, \tag{6.14}$$

$$\mathbf{F}_Q = F_{\mathrm{T}}\,\mathbf{t}_Q\,, \tag{6.15}$$

**Figure 6.5** Computation of the force $\mathbf{F}$ and the moment $\mathbf{M}$ applied to the surface coordinate frame $\mathcal{K}_\mathcal{S}$ using the tensional muscle force $F_\mathrm{T}$. The forces $\mathbf{F}_P = -F_\mathrm{T}\,\mathbf{t}_P$ and $\mathbf{F}_Q = F_\mathrm{T}\,\mathbf{t}_Q$ at the geodesic's boundary points are given by the tensional muscle force $F_\mathrm{T}$ along the muscle and the directions of the tangent vectors $\mathbf{t}_P$ and $\mathbf{t}_Q$, respectively.




where $\mathbf{t}_P$ and $\mathbf{t}_Q$ are the geodesic's tangent vectors at $P$ and $Q$, respectively. Accordingly, the resulting force $\mathbf{F}$ is given by the compact expression

$$\mathbf{F} = F_\mathrm{T}\,(\mathbf{t}_Q - \mathbf{t}_P) \tag{6.16}$$

and the moment $\mathbf{M}$ is given by

$$\mathbf{M} = \mathbf{x}_P \times \mathbf{F}_P + \mathbf{x}_Q \times \mathbf{F}_Q\,, \tag{6.17}$$

where $\mathbf{x}_P$ and $\mathbf{x}_Q$ are vectors from the surface coordinate frame $\mathcal{K}_\mathcal{S}$ to points $P$ and $Q$, respectively.

# 7 Extension to implicit surfaces

The previous chapters assume that wrapping surfaces are described parametrically. This chapter provides the necessary mathematical foundations to simulate shortest paths over implicit surfaces. It introduces explicit expressions for essential differential-geometric quantities that are necessary to compute geodesics on implicit surfaces and to establish the global path-error Jacobian. It also presents a method for computing the signed distance between a surface and the path. For a deeper study of implicit surfaces, see for instance [74, 75].

Implicit surfaces are described by a scalar equation of the form

$$f(\mathbf{p}) = 0\,, \quad \mathbf{p} := \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3\,, \quad f \in \mathbb{R}\,, \tag{7.1}$$

where any point $\mathbf{p}$ that fulfills Eqn. 7.1 is a point on the surface. An implicit surface is said to be differentiable if the function $f$ is differentiable with respect to $\mathbf{p}$. For the methods described in this work it is necessary that $f$ is two-times differentiable so that the gradient

$$\nabla f := \frac{\partial f}{\partial \mathbf{p}} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{7.2}$$

and the Hessian

$$\mathbf{H} := \frac{\partial^2 f}{\partial \mathbf{p}^2} = \begin{bmatrix} f_{xx} & f_{yx} & f_{zx} \\ f_{xy} & f_{yy} & f_{zy} \\ f_{xz} & f_{yz} & f_{zz} \end{bmatrix} \tag{7.3}$$

exist. Note that by Young's theorem it holds $f_{xy} = f_{yx}$ etc. so the Hessian is symmetric. The adjoint $\mathrm{adj}(\mathbf{H})$ of $\mathbf{H}$ is given by

$$\mathrm{adj}(\mathbf{H}) := \begin{bmatrix} f_{yy}f_{zz} - f_{yz}f_{zy} & f_{yz}f_{zx} - f_{yx}f_{zz} & f_{yx}f_{zy} - f_{yy}f_{zx} \\ f_{xz}f_{zy} - f_{xy}f_{zz} & f_{xx}f_{zz} - f_{xz}f_{zx} & f_{xy}f_{zx} - f_{xx}f_{zy} \\ f_{xy}f_{yz} - f_{xz}f_{yy} & f_{yx}f_{xz} - f_{xx}f_{yz} & f_{xx}f_{yy} - f_{xy}f_{yx} \end{bmatrix} . \tag{7.4}$$

It is possible to express all required quantities for the methods described in this work using the above expressions for the gradient $\nabla f$, the Hessian $\mathbf{H}$, and the Hessian's adjoint $\mathrm{adj}(\mathbf{H})$.

## 7.1 Normal vector and Gaußian curvature

The outward normal vector of an implicit surface is given by

$$\mathbf{N} := \frac{\nabla f}{\| \nabla f \|} . \tag{7.5}$$

The Gaußian curvature $K$ is given by the expression [75]

$$K = \frac{\nabla f \cdot (\mathrm{adj}(\mathbf{H}) \nabla f)}{\| \nabla f \|^4} , \tag{7.6}$$

where $\cdot$ again indicates a scalar product.

## 7.2 Normal curvature

According to the formulas of the general Darboux trihedron in Eqn. 3.63, it holds

$$\mathbf{N}' = -\kappa_n \, \mathbf{t} - \tau_g \, \mathbf{B} . \tag{7.7}$$

Scalar multiplication of Eqn. 7.7 with $\mathbf{t}$ gives the explicit expression

$$\kappa_n := -\mathbf{t} \cdot \mathbf{N}' , \tag{7.8}$$

where the prime indicates a derivative with respect to arc length. It is straightforward to show that

$$\nabla f' = \mathbf{H} \, \mathbf{p}' \tag{7.9}$$

and that

$$\| \nabla f \|' = \mathbf{N} \cdot (\mathbf{H}\,\mathbf{p}') \ . \tag{7.10}$$

Using the above two expressions, $\mathbf{N}'$ can be written as

$$\mathbf{N}' = \frac{\mathbf{H}\,\mathbf{p}' - \mathbf{N} \cdot [\mathbf{N} \cdot (\mathbf{H}\,\mathbf{p}')]}{\| \nabla f \|} \ . \tag{7.11}$$

Inserting Eqn. 7.11 into Eqn. 7.8 and substituting $\mathbf{t} = \mathbf{p}'$ gives

$$\kappa_n = \frac{\mathbf{p}' \cdot (\mathbf{H}\,\mathbf{p}')}{\| \nabla f \|} \ . \tag{7.12}$$

Here, $\mathbf{p}'$ is a unit vector that is tangent to the surface. It defines the direction in which the normal curvature $\kappa_n$ is computed.

## 7.3 Geodesic torsion

The expression for the geodesic torsion is derived similarly to the expression for the normal curvature. Scalar multiplication of Eqn. 7.7 with $-\mathbf{B} = \mathbf{N} \times \mathbf{t}$ yields

$$\tau_g := \mathbf{N}' \cdot (\mathbf{N} \times \mathbf{t}) \ . \tag{7.13}$$

Inserting Eqn. 7.11 in Eqn. 7.13 and substituting $\mathbf{t} = \mathbf{p}'$ gives the final formula

$$\tau_g = \frac{(\mathbf{H}\,\mathbf{p}') \cdot (\nabla f \times \mathbf{p}')}{\| \nabla f \|^2} \ , \tag{7.14}$$

where $\mathbf{p}'$ is again a unit vector that is tangent to the surface.

## 7.4 Geodesic equations

The equations for a geodesic curve on an implicit surface can be derived by considering the motion of a moving particle whose motion is constrained by the implicit surface equation

$f(\mathbf{p}) = 0$, i.e., the particle must stay on the surface. The general Lagrange equations of the first kind for the moving particle are

$$\begin{bmatrix} \mathbf{p}'' \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \nabla f \\ \nabla f^{\mathrm{T}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ -\nabla f' \cdot \mathbf{p}' \end{bmatrix}, \tag{7.15}$$

where $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the mass matrix and $\lambda$ is the so-called Lagrange multiplier related to the constraint function $f$. Because of the simple structure of the matrix in Eqn. 7.15, it can be inverted explicitly. After performing the matrix inversion and multiplying with the right hand side Eqn. 7.15 simplifies to

$$\begin{bmatrix} \mathbf{p}'' \\ \lambda \end{bmatrix} = \frac{\mathbf{p}' \cdot (\mathbf{H}\,\mathbf{p}')}{\| \nabla f \|^2} \begin{bmatrix} -\nabla f \\ 1 \end{bmatrix}. \tag{7.16}$$

Thus the geodesic equations for implicit surfaces in normal form can be written as

$$\mathbf{p}'' + \frac{\mathbf{p}' \cdot (\mathbf{H}\,\mathbf{p}')}{\| \nabla f \|^2} \nabla f = 0\,. \tag{7.17}$$

Note that Eqn. 7.17 can also be written, using Eqn. 7.12, as

$$\mathbf{p}'' = -\kappa_n \mathbf{N}\,. \tag{7.18}$$

The above equation confirms the fundamental fact that the normal vector of a geodesic is always parallel to the surface normal.

The geodesic Eqn. 7.17 is a second-order ODE that has a unique solution for a start point $\mathbf{p}_P$, a direction $\mathbf{p}'_P$, and a length $\ell_\gamma$. Thus a geodesic on an implicit surface is uniquely defined by the following seven parameters

$$\mathbf{q} := \begin{bmatrix} \mathbf{p}_P \\ \mathbf{p}'_P \\ \ell_\gamma \end{bmatrix} \in \mathbb{R}^{7\times1}\,. \tag{7.19}$$

However, these parameters are subject to constraints: point $\mathbf{q}_P$ must fulfill the on-surface constraint Eqn. 7.1, and $\mathbf{q}'_P$ must both be tangent to the surface and be a unit vector. These additional constraints are compactly written as

$$\mathbf{q}'_P \cdot \mathbf{N} = 0\,, \tag{7.20}$$

$$\| \mathbf{q}'_P \| = 1\,. \tag{7.21}$$

Fulfilling the constraints Eqn. 7.1, Eqn. 7.20, and Eqn. 7.21 at each iteration step is essential for the muscle wrapping method, in particular when correcting the geodesic parameters given a set of natural geodesic variations, as explained next.

## 7.5 Updating the geodesic parameters

With the quantities derived in this chapter, it is possible to compute all terms in the global path-error Jacobian (Eqn. 4.47) when using implicit surfaces. With the Jacobian, a set of four natural geodesic corrections $\Delta\boldsymbol{\xi}$ can be computed for each geodesic. This section describes how to map these four natural corrections to a new set of seven implicit geodesic parameters. In this setting, $k$ corresponds to the current iteration step, where $k+1$ means the subsequent step.

### 7.5.1 Start-point position correction

Let the two natural start-point corrections $(\Delta s_P, \Delta\beta_P)_{(k)}$ be given and let $\mathbf{p}_{P,(k)}$ describe the position of the current geodesic start point $P_{(k)}$. Then the vector

$$\Delta\mathbf{p}_{P,(k)} = \begin{bmatrix} \mathbf{t}_P & \mathbf{B}_P \end{bmatrix}_{(k)} \begin{bmatrix} \Delta s_P \\ \Delta\beta_P \end{bmatrix}_{(k)} \tag{7.22}$$

describes the computed displacement of the start point. Since $\Delta\mathbf{p}_{P,(k)}$ lies in the tangent plane spanned by $\{\mathbf{t}_P, \mathbf{B}_P\}_{(k)}$, the displaced point $\hat{P}_{(k+1)}$ with the position

$$\hat{\mathbf{p}}_{P,(k+1)} = \mathbf{p}_{P,(k)} + \Delta\mathbf{p}_{P,(k)} \tag{7.23}$$

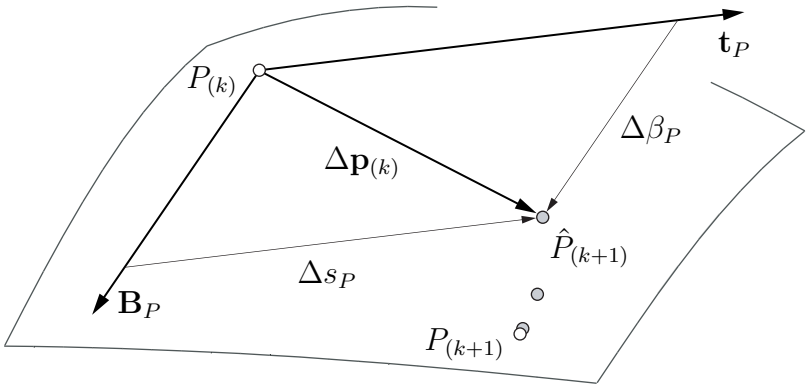
does not fulfill the implicit surface Eqn. 7.1 anymore (Fig. 7.1). Thus $\hat{\mathbf{p}}_{P,(k+1)}$ must be projected back onto the surface. This projection is performed iteratively.

For a generic point $\tilde{\mathbf{p}}$ with $f(\tilde{\mathbf{p}}) \neq 0$ the iterative projection rule reads

$$\tilde{\mathbf{p}}_{(m+1)} = \tilde{\mathbf{p}}_{(m)} - \left( \frac{\nabla f}{\| \nabla f \|^2} f \right)_{(m)}, \tag{7.24}$$

where $m$ denotes the current iteration step. This rule is executed until $f$ is sufficiently close to zero. Note that the so found point $P_{(k+1)}$ is generally not exactly the closest on-surface point to $\hat{P}_{(k+1)}$.

**Figure 7.1** Displacing the geodesic's start point $P_{(k)}$ by $\Delta\mathbf{p}_{(k)}$ according to the corrections $(\Delta s_P, \Delta\beta_P)_{(k)}$ causes $P_{(k)}$ to leave the surface. The displaced point $\hat{P}_{(k+1)}$ is iteratively projected back onto the surface such that the new geodesic start point $P_{(k+1)}$ fulfills the on-surface constraint (Eqn. 7.1).




### 7.5.2 Initial direction correction

The correction of the geodesic's initial direction according to the finite correction $\Delta\theta_{(k)}$ involves two steps. Analogously to the parametric surfaces, the first step consists in rotating the tangent $\mathbf{t}_{P,(k)}$ at point $P_{(k)}$ according to the correction $\Delta\theta_{(k)}$ using Eqn. 4.62

$$\hat{\mathbf{t}}_{P,(k)} = \mathbf{t}_{P,(k)} \cos\Delta\theta_{(k)} + \mathbf{B}_{P,(k)} \sin\Delta\theta_{(k)} \,. \tag{7.25}$$

The resulting vector $\hat{\mathbf{t}}_{P,(k)}$ is then parallel transported to the new start point $P_{(k+1)}$, where it is normalized again to fulfill both the tangency constraint (Eqn. 7.20) and the unit-speed constraint (Eqn. 7.21). Both constraints are fulfilled after substracting the normal component of $\hat{\mathbf{t}}_{P,(k)}$ at $P_{(k+1)}$ and normalizing the vector. The new initial direction is given by

$$\mathbf{p}_{P,(k+1)} = \frac{\hat{\mathbf{t}}_{P,(k)} - \left(\hat{\mathbf{t}}_{P,(k)} \cdot \mathbf{N}_{P,(k+1)}\right) \mathbf{N}_{P,(k+1)}}{\| \hat{\mathbf{t}}_{P,(k)} - \left(\hat{\mathbf{t}}_{P,(k)} \cdot \mathbf{N}_{P,(k+1)}\right) \mathbf{N}_{P,(k+1)} \|} \,. \tag{7.26}$$

### 7.5.3 Length correction

The correction of the geodesic's length does not depend on the way the wrapping surface is described. Thus Eqn. 4.67 holds for implicit surfaces as well. It is rewritten here for the sake of lucidity

$$\ell_{\gamma,(k+1)} = \ell_{\gamma,(k)} + \Delta\ell_{\gamma,(k)} \,. \tag{7.27}$$

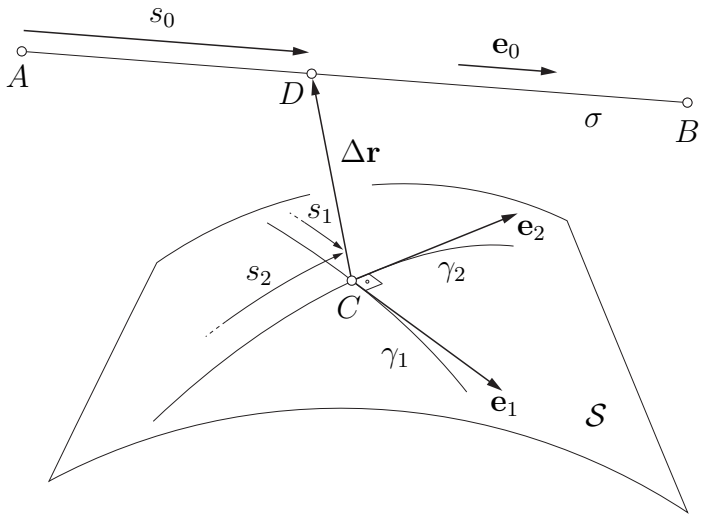
## 7.6 Signed distance between path and surface

The computation of the signed distance between a straight-line segment and an implicit surface is performed similarly to the computation for parametric surfaces, described in Sec. 6.1. The major difference lies in the fact that on an implicit surface there is no tangential basis available right away for formulating a set of orthogonality constraints. Such a basis needs to be established first, as described next.

Following the derivations for parametric surfaces in Sec. 6.1, $\sigma$ is a straight-line segment between two generic points $A$ and $B$, $\mathbf{e}_0$ is the unit vector along $\sigma$, $s_0$ parameterizes the position of a point $D$ on $\sigma$, and $C$ is a point on the surface (Fig. 7.2). The goal is to find the positions of points $C$ and $D$ such that the connecting vector $\Delta\mathbf{r} = \mathbf{r}_D - \mathbf{r}_C$ is both orthogonal to the straight-line segment and to the surface.

Measuring if $\Delta\mathbf{r}$ is orthogonal to the surface requires an arbitrary tangential basis at point $C$. Such a basis can easily be established using the surface normal and one arbitrary non-parallel vector, thus a further description is left out here.

**Figure 7.2** Computation of the signed distance between a path segment $\sigma$ and an implicit surface $\mathcal{S}$. Sought are two points $C$ on $\mathcal{S}$ and $D$ on $\sigma$ such that the resulting vector $\Delta\mathbf{r} = \mathbf{r}_D - \mathbf{r}_C$ is both orthogonal to $\mathcal{S}$ and to $\sigma$. The deviation from orthogonality is measured using the unit vector $\mathbf{e}_0$ along $\sigma$ and a tangential orthogonal basis $\{\mathbf{e}_1, \mathbf{e}_2\}$ on $\mathcal{S}$. The position of $D$ is parameterized by its distance $s_0$ from the straight-line segment's start point $A$, while the position of $C$ is assumed to be parameterized by the arc lengths $s_1$ and $s_2$ along two orthogonally intersecting geodesics $\gamma_1$ and $\gamma_2$ at $C$.



Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be two orthogonal unit vectors that span the tangent plane at point $C$, defined such that $\mathbf{e}_1 \times \mathbf{e}_2 = \mathbf{N}_C$. Then the points $C$ and $D$ are sought such that the following constraint equation is fulfilled

$$\mathbf{g}(\mathbf{w}) := \begin{bmatrix} \mathbf{e}_0 & \cdot & \Delta\mathbf{r} \\ \mathbf{e}_1 & \cdot & \Delta\mathbf{r} \\ \mathbf{e}_2 & \cdot & \Delta\mathbf{r} \end{bmatrix} = \mathbf{0}\,, \quad \mathbf{w} := \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix}. \tag{7.28}$$

Here $s_1$ and $s_2$ can be regarded as arc lengths along two orthogonally intersecting geodesics $\gamma_1$ and $\gamma_2$ through point $C$ of which $\mathbf{e}_1$ and $\mathbf{e}_2$ are the respective tangent vectors. The geodesics $\gamma_1$ and $\gamma_2$ form an orthogonal coordinate system, originally introduced by Gauß, with $s_1$ and $s_2$ being the respective coordinates.

The goal is to use the explicit gradient $\partial\mathbf{g}/\partial\mathbf{w}$ to zero $\mathbf{g}$. Note that $\{\mathbf{e}_2, \mathbf{N_C}, \mathbf{e}_1\}$ can be regarded as an orthogonal Darboux trihedron so the previously derived equations Eqn. 4.12 and Eqn. 4.13 for the Darboux trihedron derivatives $\partial\mathcal{K}_P/\partial s_P$ and $\partial\mathcal{K}_P/\partial\beta_P$ hold. Analog to Eqn. 4.12 and Eqn. 4.13 it holds for the derivatives of $\mathbf{e}_1$ and $\mathbf{e}_2$ with respect to the arc lengths $s_1$ and $s_2$

$$\frac{\partial\mathbf{e}_1}{\partial s_1} = \kappa_n^{\gamma^1}\,\mathbf{N}_C\,, \tag{7.29}$$

$$\frac{\partial\mathbf{e}_1}{\partial s_2} = -\tau_g^{\gamma^2}\,\mathbf{N}_C\,, \tag{7.30}$$

$$\frac{\partial\mathbf{e}_2}{\partial s_1} = \tau_g^{\gamma^1}\,\mathbf{N}_C\,, \tag{7.31}$$

$$\frac{\partial\mathbf{e}_2}{\partial s_2} = \kappa_n^{\gamma^2}\,\mathbf{N}_C\,. \tag{7.32}$$

With the above equations and the relationship $-\tau_g^{\gamma^2} = \tau_g^{\gamma^1}$ the respective Jacobian $\partial\mathbf{g}/\partial\mathbf{w}$ takes the form

$$\frac{\partial\mathbf{g}}{\partial\mathbf{w}} = \begin{bmatrix} 1 & -\mathbf{e}_0\cdot\mathbf{e}_1 & -\mathbf{e}_0\cdot\mathbf{e}_2 \\ \mathbf{e}_0\cdot\mathbf{e}_1 & \kappa_n^{\gamma^1}\,\mathbf{N}_C\cdot\Delta\mathbf{r} - 1 & \tau_g^{\gamma^1}\,\mathbf{N}_C\cdot\Delta\mathbf{r} \\ \mathbf{e}_0\cdot\mathbf{e}_2 & \tau_g^{\gamma^1}\,\mathbf{N}_C\cdot\Delta\mathbf{r} & \kappa_n^{\gamma^2}\,\mathbf{N}_C\cdot\Delta\mathbf{r} - 1 \end{bmatrix}. \tag{7.33}$$

The Jacobian allows for computing a set of corrections

$$\Delta\mathbf{w} := \begin{bmatrix} \Delta s_0 \\ \Delta s_1 \\ \Delta s_2 \end{bmatrix} = \left(\frac{\partial\mathbf{g}}{\partial\mathbf{w}}\right)^{-1}\,\mathbf{g}\,, \tag{7.34}$$

which can be used to iteratively correct the positions of points $C$ and $D$ until $\| \mathbf{g} \|$ is sufficiently close to zero.

However, the computation of a new position of point $C$ is different from the equivalent computation on parametric surfaces because there are no actual coordinates on implicit surfaces available. As a result, it is not possible to compute a new set of coordinates for point $C$ by just adding the computed corrections $\Delta s_1$ and $\Delta s_2$ to the previous values of $s_1$ and $s_2$. A reasonable solution can be found by first displacing point $C$ in the tangent plane and then projecting it back to the surface, as described in Sec. 7.5.1. The following example illustrates the above explained computation of the signed distance between a straight-line segment and an implicit ellipsoid.

### 7.6.1 Example

This example illustrates the iterative computation of the signed distance between a straight-line segment and an implicit ellipsoid. Let the positions of points $A$ and $B$ with respect to the ellipsoid be given by $\mathbf{r}_A = \begin{bmatrix} -10 & 10 & 0 \end{bmatrix}$ and $\mathbf{r}_B = \begin{bmatrix} 10 & 5 & 3 \end{bmatrix}$ and let the ellipsoid be defined by $f = x^2/25 + y^2/9 + z^2/16 = 0$. Figure 7.3 shows the setup as well as the positions of points $C$ and $D$ during the iterations.

**Figure 7.3** Iterative computation of the signed distance between a straight-line segment and an ellipsoid. Shown are the initial guess $C_{(1)}$ and $D_{(1)}$, the first iteration $C_{(2)}$ and $D_{(2)}$, and the solution $C_{(6)}$ and $D_{(6)}$. Note that the iterations $C_{(3)}$ to $C_{(5)}$ and $D_{(3)}$ to $D_{(5)}$ are not visible in the figure due to their proximity to $C_{(6)}$ and $D_{(6)}$, respectively.

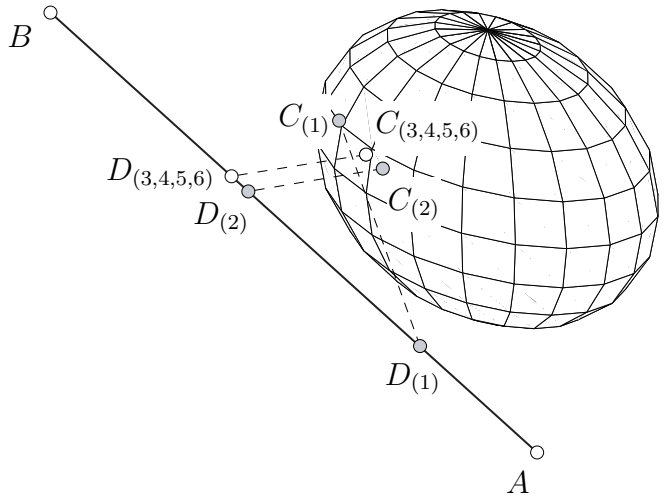


**Table 7.1** Numerical results for the signed distance computation example. The positions of point $C$ on the surface and point $D$ on the straight-line segment were computed such that the distance between $C$ and $D$ is minimal. The history of the error $\| \mathbf{g} \|$ shows quadratic convergence near the solution, allowing for computing high-precision solutions in a few iterations.

| iter. $k$ | base point (surf.) $C_{(k)}$ | base point (line) $D_{(k)}$ | error norm $\| \mathbf{g} \|$ | sign. dist. $d$ |
|---|---|---|---|---|
| 1 | $\begin{bmatrix} 2.95 & 2.22 & 1.28 \end{bmatrix}$ | $\begin{bmatrix} -5.20 & 8.80 & 0.72 \end{bmatrix}$ | 13.980143866214055 | 10.49 |
| 2 | $\begin{bmatrix} 1.00 & 2.86 & 0.91 \end{bmatrix}$ | $\begin{bmatrix} 1.87 & 7.03 & 1.78 \end{bmatrix}$ | 0.367150387900821 | 4.35 |
| 3 | $\begin{bmatrix} 1.67 & 2.72 & 1.02 \end{bmatrix}$ | $\begin{bmatrix} 2.57 & 6.86 & 1.89 \end{bmatrix}$ | 0.009824863697499 | 4.32 |
| 4 | $\begin{bmatrix} 1.65 & 2.73 & 1.02 \end{bmatrix}$ | $\begin{bmatrix} 2.56 & 6.86 & 1.88 \end{bmatrix}$ | 0.000011270733715 | 4.32 |
| 5 | $\begin{bmatrix} 1.65 & 2.73 & 1.02 \end{bmatrix}$ | $\begin{bmatrix} 2.56 & 6.86 & 1.88 \end{bmatrix}$ | 0.000000000014995 | 4.32 |
| 6 | $\begin{bmatrix} 1.65 & 2.73 & 1.02 \end{bmatrix}$ | $\begin{bmatrix} 2.56 & 6.86 & 1.88 \end{bmatrix}$ | 0.000000000000000 | 4.32 |

The initial guesses for points $C$ and $D$ were chosen such that the initial error norm was $\| \mathbf{g} \| = 13.98$. At the sixth iteration step the error norm was successfully zeroed to a value near machine precision, i.e., $\| \mathbf{g} \| = 2.22\text{E-}16$. At this solution, the signed distance between the surface and the line was 4.32. Table 7.1 shows the results for the six iteration steps, confirming the expected quadratic convergence of the iterations near the solution.

# 8 Implementation and application examples

This section contains (*i*) details about the implementation of the method into a multibody simulation software package (Sec. 8.1), and (*ii*) the results of four simulation examples that were designed to assess the proposed method's numerical properties and its general applicability to biomechanical problems (Sec. 8.2 – Sec. 8.5).

## 8.1 Implementation

The method has been implemented into the C++ multibody simulation environment Mobile [76]. Despite the extensive use of differential geometry, the method's algorithmic implementation is surprisingly straightforward. It comprises

1. a set of smooth surface objects, each capable of calculating elementary differential-geometric entities such as the surface normal vector, normal curvature, geodesic torsion, and Gaußian curvature. Surface objects can be either described parametrically or implicitly, depending on the user's preferences;
2. a geodesic shooter that computes one geodesic and the two associated Jacobi fields on each surface given the geodesic's start point, direction, and length. The shooter also computes the Darboux trihedra and their derivatives at the geodesic's boundary points with respect to the four natural geodesic variations;
3. a global equation assembler that computes (*i*) the straight-line segments to evaluate the path error using the Darboux trihedra, and (*ii*) the banded path-error Jacobian explicitly using the derivatives of the Darboux trihedra and the derivatives of the straight-line unit vectors;
4. an iterative solver which computes a set of four natural geodesic corrections per geodesic to correct the parameters of all geodesics until the path error vanishes and the shortest path is found;
5. an explicit evaluator for the path's rate of length change; and
6. an iterative solver which computes the signed distances of all wrapping surfaces to the path to detect path lift-off and touchdown.

Note from the items above that the computational costs of the proposed method are largely determined by (*i*) integrating the geodesic equations and (*ii*) solving the linear

equation system for computing the natural geodesic corrections. For the example simulations shown in this chapter, ODEPACK's [77] variable step-size integrator LSODAR was used for shooting geodesics, and LAPACKS's [78] band matrix routine DGBSV was used for computing the geodesic corrections. The examples were simulated on a normal desktop computer with an Intel(R) Core(TM) i7-2700K @3.50GHz, 8GB RAM, running Windows 7 Professional, 64 bit. In all example simulations, Newton's method was used to iteratively correct the geodesic parameters. In this setting, the solution parameters of a previous time step were used as the initial guess for the current time step, with manually determined initial conditions at the start (Cha. 9).

The first example simulation (Sec. 8.2) highlights numerical aspects of the signed distance computation algorithm during a continuous simulation. The second example (Sec. 8.3) shows a dynamic simulation of a single muscle path across multiple nonsimple wrapping surfaces, where two surfaces are described parametrically and the other two surfaces are described implicitly. The third example (Sec. 8.4) demonstrates the simulation of a sheet muscle, i.e., the latissimus dorsi that is discretized by multiple shortest-paths, across a parametric surface patch that is fitted to the human ribcage. The fourth and last example (Sec. 8.5) compares the method's computational costs when simulating a single path over a variable number of wrapping surfaces. This example also compares the computational costs when using implicit surfaces instead of parametric surfaces.

## 8.2 Path lift-off and touchdown

This example shows the numerical behavior of the contact detection algorithm and the time histories of the signed distance as well as of path length and rate of length change. The example consists of a parametric paraboloid (Fig. 8.1) that performs sinosoidal vertical motions and a single muscle path with fixed origin and insertion points. The paraboloid starts moving at its lowest position. It makes contact with the path while moving up, and breaks contact with the path while moving down. The total simulation time is 5.0 s and lengths are measured in a generic length unit LU.

**Figure 8.1** Example simulation showing path lift-off and touchdown. A vertically moving paraboloid makes contact with the path while moving up and breaks contact with the path while moving down.

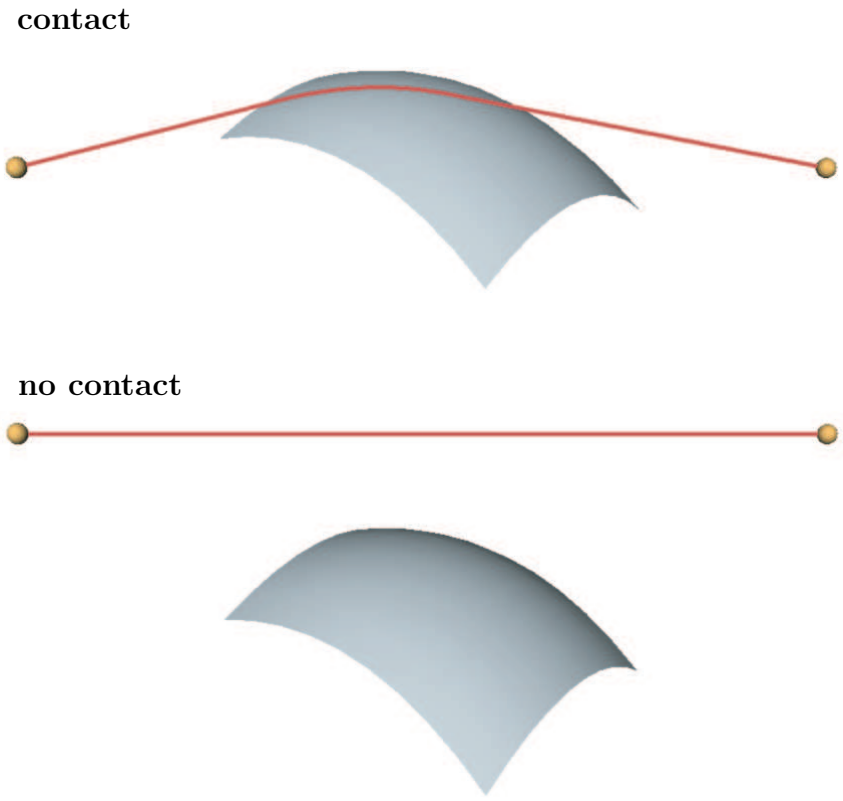
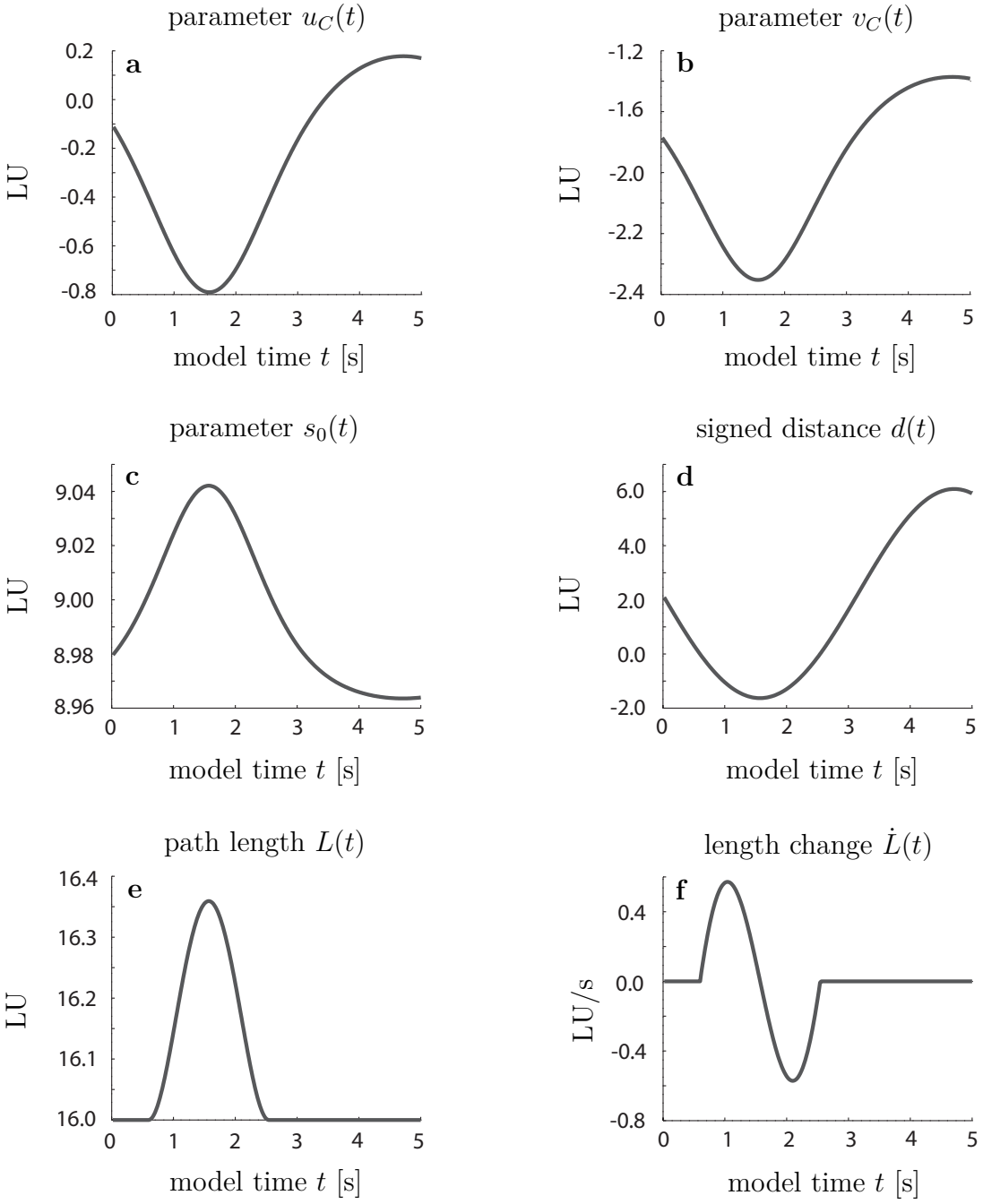



Figure 8.2d shows a plot of the time history of the signed distance $d$ during the simulation. The plot shows that $d$ is well-suited to identify contacts between shortest paths and obstacle surfaces because it is $(i)$ smooth and $(ii)$ crosses $d = 0$ continuously. Figure 8.2a and Fig. 8.2b show the time histories of the surface coordinates of point $C$ on the surface, which is one of two base points for the signed-distance computation (Fig. 6.1). Figure 8.2c shows the time history of the straight-line coordinate $s_0$ of point $D$, which is the other base point. Figure 8.2e and Fig. 8.2f show plots of path length and rate of length change.

**Figure 8.2** Time histories of **(a, b)** the coordinates $u_C$ and $v_C$ of the surface base point $C$; **(c)** the parameter $s_0$ describing the position of point $D$ on the straight-line segment; **(d)** the signed distance $d$ between the paraboloid and the straight-line segment; **(e)** the path length $L$; and **(f)** the path rate of length change $\dot{L}$. Lengths are measured a generic length unit (LU), while the time $t$ is measured in seconds.

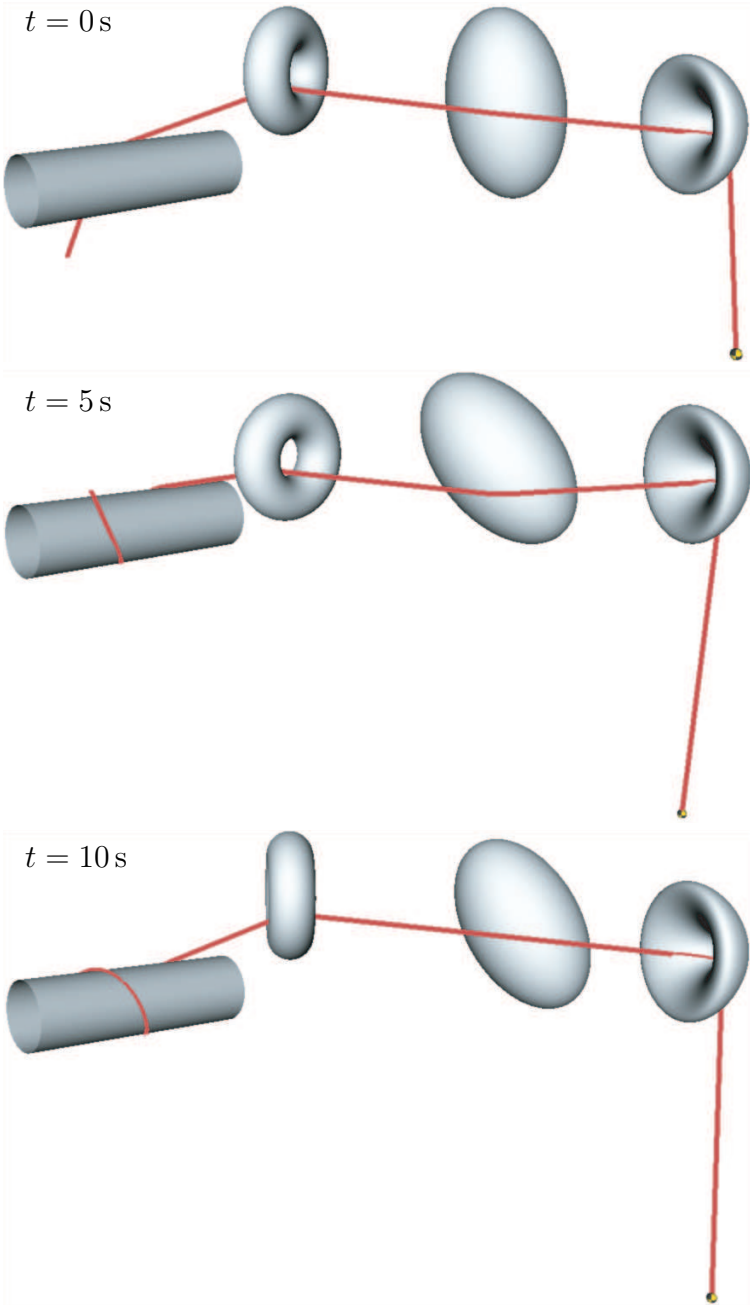


## 8.3 Dynamic simulation

This example is a force-driven simulation of a single muscle path over four (nonsimple) wrapping surfaces: a cylinder, a torus, an ellipsoid, and an elliptic torus (Fig. 8.3). A freely moving point mass is attached to one end point of the path, which is assumed to act as a linear elastic string that generates tensional force when being pulled beyond is slack length. The other end of the path is wrapped around the cylinder during simulation. To impose motion to the system, the surfaces are kinematically driven and perform sinusoidal spatial translations and rotations. In this setting, the ellipsoid makes and breaks contact with the path during simulation.

For comparing implicit and parametric surfaces, the cylinder and the ellipsoid are either described implicitly or parametrically, while the torus and the elliptic torus are described parametrically in both cases. The goal of this example simulation is threefold: (*i*) to show the convergence behavior of the path corrections given a far away initial guess for the geodesic parameters prior to simulation start; (*ii*) to show that the method can work with multiple nonsimple surfaces that can be either described parametrically or implicitly, where both representations yield equal results; and (*iii*) to show that the method works in a force-driven simulation, including path lift-off and touchdown. Figure 8.3 shows the example setup at its initial configuration at $t = 0\,\mathrm{s}$ as well as at $t = 5\,\mathrm{s}$ and $t = 10\,\mathrm{s}$.

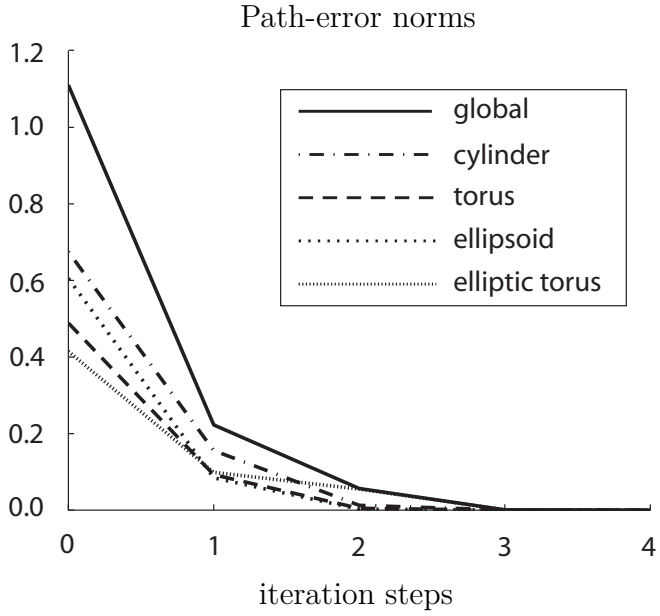
**Figure 8.3** Wrapping over an implicit cylinder, a parametric torus, an implicit ellipsoid, and an elliptic torus. The shortest path is modeled as an elastic band at whose free end a point mass is attached. The figure shows three recordings of the simulation at $t = 0\,\mathrm{s}$, $t = 5\,\mathrm{s}$, and $t = 10\,\mathrm{s}$.




Prior to the simulation, a set of geodesic parameters was chosen manually, which caused an initial path-error norm of $\| \varepsilon \| = 1.11$. To start the simulation with a feasible set of geodesic parameters, five path iterations were executed until $\| \varepsilon \| = 8.13\text{E-}13$. Figure 8.4 shows the convergence behavior of the local path errors on the surfaces and of the global path error during the first four iterations. It is clearly visible from Fig. 8.4 that all path

errors converge quadratically to zero, allowing for the computation of highly accurate solutions in a few iteration steps.

**Figure 8.4** Path-error norms of the obstacles during the first four iterations at initialization. Prior to the simulation, the path error norm was $\| \varepsilon \| = 1.11$. The individual local path error norms converge quadratically to zero near the solution. After five iterations, the path error was 8.13E-13.




For the actual simulation, the tolerance for the path-error norm was set to 1.0E-6. During the simulation, the geodesic parameters of the previous time step were used as the initial guess for the current time step. Since the time steps of the integrator are small, it can be assumed that subsequent sets of geodesic parameters are coherent and thereby previous solutions are sufficiently close initial guesses.

**Figure 8.5** Time histories of path length $L$, rate of length change $\dot{L}$, and global path-error norm $\|\varepsilon\|$ during 10 s of simulation. Panel **(a)** shows the path length and the rate of length change when the cylinder and the ellipsoid are described implicitly. Panel **(b)** shows the same quantities when the cylinder and the ellipsoid are described parametrically. Analogously, panels **(c)** and **(d)** show the global path error norms for both cases, respectively.

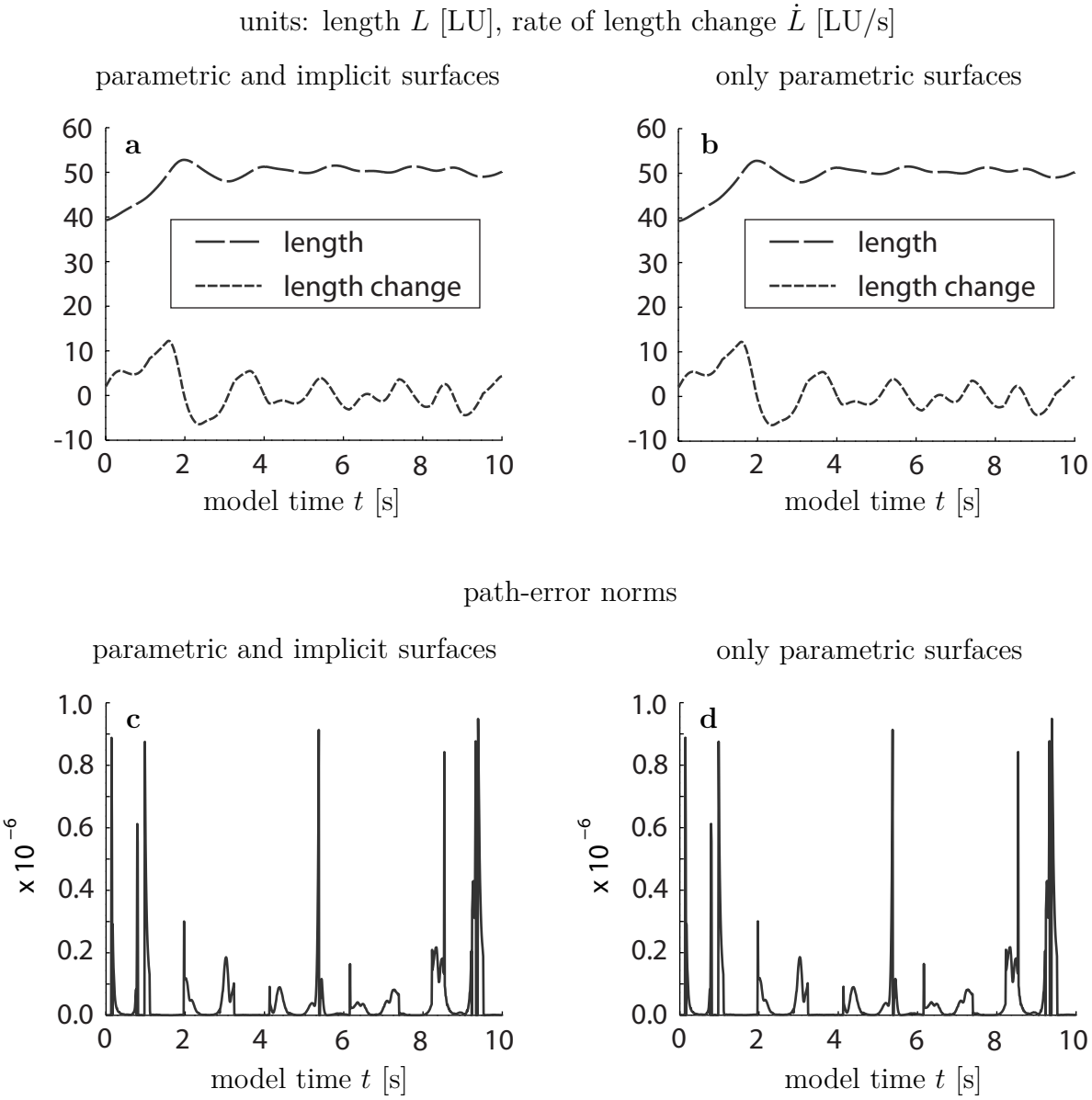



Figure 8.5 shows the time histories of path length $L$, rate of length change $\dot{L}$, and global path-error norm $\|\varepsilon\|$ during the first 10 s of the simulation, both when using only parametric surfaces and when also using implicit surfaces. These plots demonstrate that the results obtained when using implicit surfaces are nearly identical to the results obtained when using parametric surfaces. When parametric and implicit surfaces were used for the simulation, the maximum path error norm was $\|\varepsilon\| = 9.52\text{E-}7$ and the

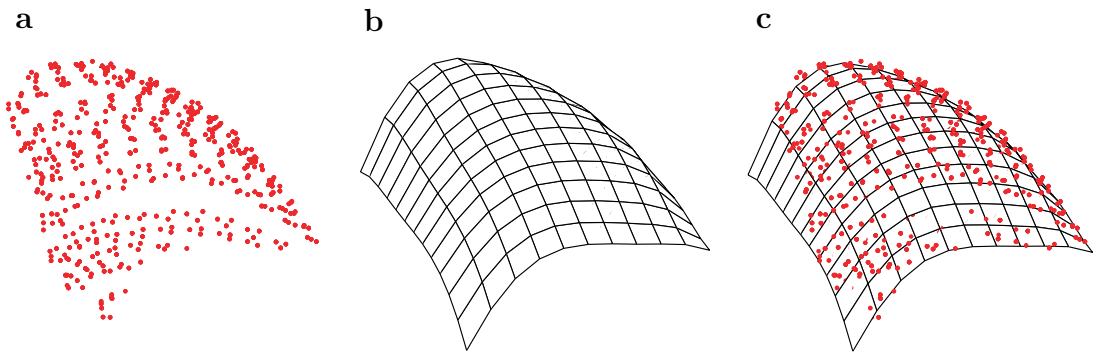
average was 3.56E-8. When only parametric surfaces were used, the maximum path error was 9.48E-7 and the average was 3.28E-8.

## 8.4 Application to biomechanical problems

The goal of this example is to demonstrate the applicability of the proposed wrapping method to biomechanical problems. It shows the simulation of the right latissimus dorsi as it wraps over a parametric surface patch fitted to the ribcage. Though the model presented here is relatively simple, it demonstrates the method's ability to simulate musculotendon paths over biologically accurate surfaces without loss of generality.

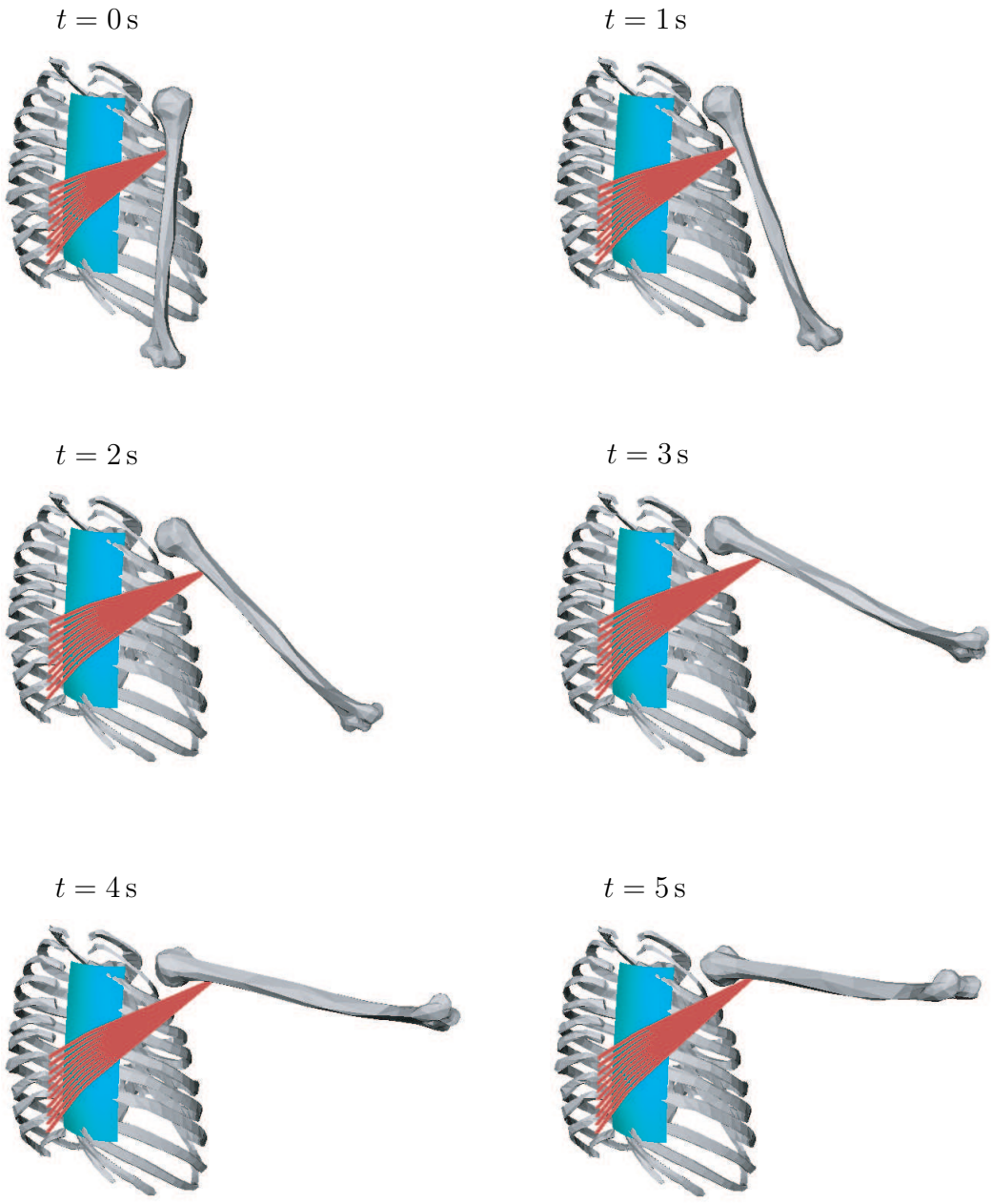
The latissimus dorsi was represented by 10 shortest paths that originate in the region of the lumbar / thoracic spine, with their insertion points being located on the humerus near the shoulder. Multiple paths were necessary to account for the sheet-muscle structure of the latissimus dorsi. A parametric surface patch was fitted to the ribcage where the muscle wraps over it. Figure 8.6 shows the raw-data point cloud that was used for surface fitting, as well as the fifth order polynomial surface patch that was computed in Matlab. For this example simulation, the skeleton consists of the ribcage and the right humerus, and the shoulder joint is modeled by a spherical joint. The geometries of ribcage and humerus are taken from OpenSim [79].

**Figure 8.6** **(a)** Point cloud of the ribcage; **(b)** surface patch computed in Matlab; **(c)** point cloud and surface patch together.



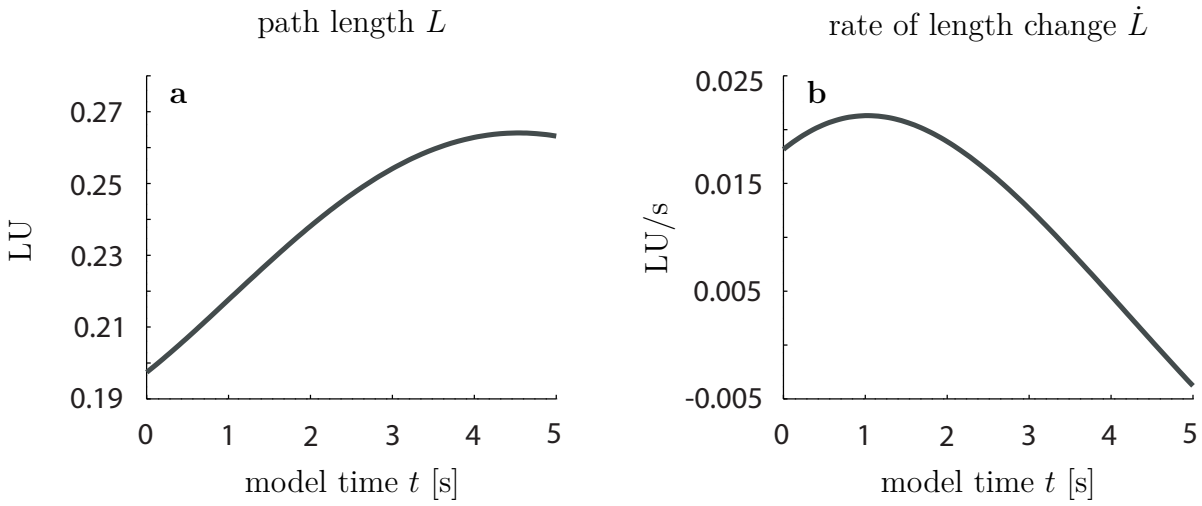

The humerus was kinematically driven for five seconds with constant angular rates about two of three joint axes. At the initial pose ($t = 0\,\mathrm{s}$), it was oriented verically, where at the final pose ($t = 5\,\mathrm{s}$) it was oriented horizontally. Figure 8.7 shows six recordings of the model. Note that for the sake of better visibility, the recordings in Fig. 8.7 show a smaller section of the parametric surface patch than Fig. 8.6. Also, the ribcage was narrowed slightly in Fig. 8.7.

**Figure 8.7** Recordings of the simple latissimus dorsi model at six different time frames.




During simulation, the length and the rate of length change of the latissimus dorsi was computed by averaging the lengths and the rates of length change of its individual shortest paths. Figure 8.8 shows the resulting time histories of path length and rate of length change during the simulation.

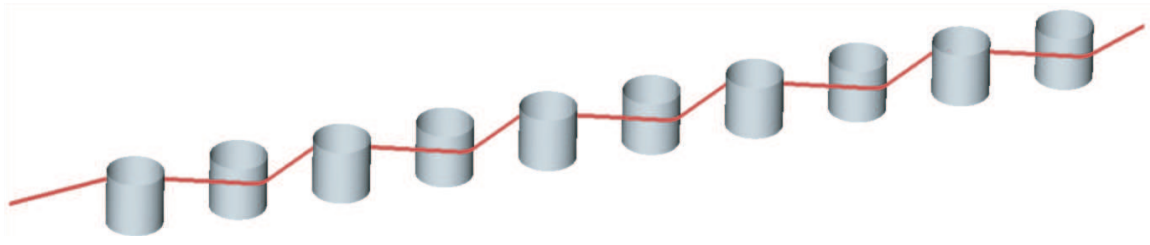
**Figure 8.8** Time histories of **(a)** path length and **(b)** rate of length change of the latissimus dorsi during 5 s of the simulation. The values were obtained by averaging the path lengths and the rates of length change of the 10 individual paths that represent the muscle.




## 8.5 Computational speed benchmark

This benchmark evaluates the method's computational speed with respect to the number of wrapping surfaces and compares the computational costs when using parametric surfaces with the computational costs when using implicit surfaces. To this end, a single path was wrapped over a variable number of cylinders that performed spatial sinusoidal translations and rotations while the path's origin and insertion points were fixed. Figure 8.9 shows the benchmark setup for $n = 10$ cylinders.

**Figure 8.9** Wrapping over a variable number of cylinders. The presented method computes high-precision solutions for shortest paths over a large number of wrapping surfaces in real time.



For this benchmark, the absolute and relative tolerances for shooting geodesics were set to 1.0E-3 and the path-error norm tolerance was set to 1.0E-3. The total simulation time was 40.0 s and the shortest path was computed every 0.02 s. Table 8.1 displays the compu-

tational costs in terms of the real-time factor when using parametric and implicit surface representations. It also shows the maximum path-error norm during the simulations.

**Table 8.1** Computational costs vs. the number of wrapping surfaces when using parametric and implicit surface representations.

| | parametric cylinders | | implicit cylinders | |
|---|---|---|---|---|
| num. cyl. | real-time factor | max. error norm | real-time factor | max. error norm |
| 100 | 0.16 | 9.5E-4 | 0.17 | 9.5E-4 |
| 200 | 0.32 | 9.3E-4 | 0.35 | 9.3E-4 |
| 300 | 0.48 | 9.9E-4 | 0.54 | 9.9E-4 |
| 400 | 0.65 | 9.2E-4 | 0.70 | 9.7E-4 |
| 500 | 0.82 | 8.5E-4 | 0.88 | 9.8E-4 |

The results in Tab. 8.1 show that (*i*) the method's computational costs grow linearly with the number of surfaces as expected; (*ii*) parametric and implicit surfaces work similarly fast; and (*iii*) the method allows for wrapping over a large number of surfaces in real time. It shall be noted here, however, that the results in Tab. 8.1 only allow for a qualitative assessment of the method's computational speed because they depend on many influencing factors such as computer hardware and software, operating system or the numerical integrator used for shooting geodesics. Nevertheless, the benchmark confirms that the proposed method is computationally very efficient and allows for the fast computation of musculotendon paths over many wrapping surfaces.

# 9 Conclusions and outlook

Muscle wrapping methods are an essential component of musculoskeletal models. Their purpose is the realistic prediction of muscle paths, their lengths, and their rates of length change, for computing muscle moment arms and muscle forces (Sec. 2.4.2).

It is often necessary that musculoskeletal models can be evaluated on normal desktop computers. To achieve this goal, muscles are commonly simplified as massless taut strings that wrap frictionlessly over obstacle surfaces representing bone and tissue. Thereby, the muscle-path computation problem is reduced to one of finding the shortest path between origin and insertion.

There exist muscle wrapping methods that are sufficiently fast, yet these methods can only work with a limited number of simple surfaces that do not accurately represent their complex biologic counterparts (Sec. 2.4.3). The few methods that can work with general surfaces rely on computationally expensive formulations and thus have a significant impact on the computational costs of musculoskeletal models. As a result, biomechanists face a trade-off between the computational speed and the accuracy of their models.

This dissertation closes the gap between the fast but nongeneral and the general but slow muscle wrapping methods and thereby allows biomechanists to develop more accurate musculoskeletal models. It provides a new method that allows for the very efficient simulation of locally shortest musculotendon paths across multiple biologically realistic surfaces. In this setting, the total path is regarded as a concatenation of straight-line segments and geodesic segments, and the shortest-path problem is formulated as one of finding the root of a global path-error constraint equation that enforces that adjacent segments connect collinearly. The path-error constraint is zeroed iteratively using the explicit banded path-error Jacobian that maps natural variations of the geodesic parameters to path-error variations. Thereby, high computational efficiency is achieved, resulting in a minimal computational footprint in musculoskeletal models.

By its general formulation, the proposed method does not rely on special types of surfaces. It can work with general nonsimple surfaces that accurately represent their complex biological counterparts, such as surface patches fitted to measured bone geometry. In particular, the surfaces can be either described parametrically or implicitly, where multiple parametric and implicit surfaces can be wrapped simultaneously by one path.

In addition to presenting a solution to the shortest-path problem, this work also presents an explicit formula for the path's exact rate of length change, which is an important input when predicting muscle force using Hill-type muscle models. That formula is independent of the method used for computing the path and makes a numerical approximation of the path's rate of length change unnecessary.

In an attempt to provide a robust muscle wrapping method, this work also addresses the simulation of path lift-off and touchdown. To detect contact events, the signed distance between the path and a wrapping surface is used as a continuous witness function. The signed distance has proven as a reliable metric for detecting whether an obstacle is in contact with the shortest path or not.

## 9.1 Contributions

In summary, the main contributions of this thesis are:

- general formulation of the shortest-path problem as a root-finding problem whose solution does not require nested loops for finding geodesics between two points;
- efficient solution of the shortest-path problem by iteratively finding the root of a global path-error constraint equation with an explicit banded Jacobian;
- ability to work concurrently with general parametric and implicit wrapping surfaces;
- explicit determination of the path's rate of length change; and
- accurate detection of path lift-off and touchdown.

The avoidance of nested loops for finding geodesics between two points is achieved by naturally parameterizing each geodesic segment by its start point, direction, and length. Thus it suffices to compute geodesics on the surfaces only once per path iteration step, which contributes significantly to the method's efficiency.

## 9.2 Limitations

The method's major limitations result (*i*) from the underlying simplifications and (*ii*) from the method's solution approach itself.

The assumption that muscles and tendons behave like one-dimensional massless strings that take the locally shortest path from origin to insertion implies three major limitations. First, this premise makes it necessary to lump the masses of muscles with body segments, which causes errors when simulating the dynamics of musculoskeletal systems, as shown by Pai [53]. Second, the neglection of muscle thickness may result in computed moment

arms that are too short. While it is straightforward to obtain biologically accurate wrapping surfaces for thin muscles and tendons from measured bone geometry, special care is necessary when modeling the paths of thick muscles. This is because the centroid lines of thick muscles have a variable distance to underlying wrapping structures, which complicates an adequate choice of wrapping surfaces for such muscles. Third, the shortest-path assumption does not allow for representing connective tissue, thus the relative motion of computed muscle paths on wrapping surfaces may be larger than expected. Despite their limitations, however, curved-line muscles are used today in the majority of models. Even though they are relatively simple, they can adequately recreate experimentally measured moment arms, as shown, for instance, by Gatti et al. [54].

Since the proposed method is based on Newton's method, it requires an initialization before simulation start to find a feasible initial set of geodesics such that the resulting candidate path converges to the shortest path. For the benchmark simulations shown in this work, the initialization was performed manually, which is a straightforward process. However, a manual determination of geodesic parameters requires knowledge about the surface representations, which may degrade the ease of use for some users.

According to the use of Newton's method, computed muscle paths are locally length-minimizing, but no information about their global optimality is provided. This limitation, however, does not restrict the method's use in musculoskeletal models. Quite the contrary, the local convergence together with the feedback of previous solutions ensures that simulated muscle paths evolve continuously as the musculoskeletal system moves.

## 9.3 Future work

The aforementioned limitations expose a solid basis for future work, which could aim at further improving computational speed, ease of use, and model complexity.

From a numerical viewpoint, there is still room for improving the method's computational speed. It can be expected that parallelizing the computation of geodesics on multi-core processors as well as using a quasi-Newton method for the path iterations will increase the method's performance further.

From a user's viewpoint, an automated determination of an initial guess of geodesic parameters prior to simulation start remains a desirable future extension. A good initialization routine would automatically generate a feasible set of parameters for each surface geodesic. This would increase the method's ease of use by making the task of finding a feasible set of geodesic parameters independent from the surface representations.

With a view to higher model complexity, future work could be directed at incorporating muscle thickness. Thick muscle paths could render different muscle fiber lengths and thus

allow for a more realistic computation of muscle force. In this setting, it remains an open task to define meaningful thickness profiles that relate a muscle's thickness with its current length and its current stimulus. Another possible aspect for future improvements in the direction of higher model complexity is the incorporation of muscle mass to avoid errors that occur when lumping muscle mass with body segments. However, considering muscle-mass distributions along muscles would prohibit a purely geometric solution of the muscle wrapping problem using the shortest-path assumption, which has shown to be the key to efficiency. It remains to study whether leaving this assumption to incorporate muscle mass is worth the extra computational costs.

# Appendix

## List of publications and research activities

Many of the contributions and concepts presented in this thesis have been previously presented in oral presentations and publications. The following list summarizes previous publications and research activities.

### Journal publications

- **A. Scholz**, M. Sherman, I. Stavness, S. Delp, A. Kecskeméthy. A fast multi-obstacle muscle wrapping method using natural geodesic variations. Multibody System Dynamics, 36(2):195-219, 2016.

### Conference publications and proceedings

- **A. Scholz**, M. Sherman, I. Stavness, S. Delp, A. Kecskeméthy. Simulating shortest musculotendon paths across multiple biologically accurate wrapping surfaces in real time. International Society of Biomechanics, Glasgow, Scotland, July 12-16, 2015.
- **A. Scholz**, M. Sherman, I. Stavness, S. Delp, A. Kecskeméthy. Computation of shortest musculotendon paths using natural geodesic variations. ECCOMAS Multibody Dynamics, Barcelona, Spain, June 29 - July 02, 2015.
- **A. Scholz**, I. Stavness, M. Sherman, S. Delp, A. Kecskeméthy. Computing musculotendon paths and their velocities across multiple moving surfaces using Jacobi fields. ECCOMAS Multibody Dynamics, Zagreb, Croatia, July 01-04, 2013.
- **A. Scholz**, I. Stavness, M. Sherman, S. Delp, A. Kecskeméthy. Improved muscle wrapping algorithms using explicit path-error Jacobians. Computational Kinematics, Barcelona, Spain, May 12-15, 2013, pp. 395-403, edited by F. Thomas, A. P. Gracia, Springer, 2013.
- **A. Scholz**, I. Stavness, M. Sherman, S. Delp, A. Kecskeméthy. Analytical determination of path-error Jacobians using Jacobi fields for muscle wrapping across multiple moving surfaces. GAMM, Novi Sad, Serbia, March 18-22, 2013.

- **A. Scholz**, F. Geu Flores, A. Kecskeméthy. Trajectory planning optimization of mechanisms with redundant kinematics for manufacturing processes with constant tool speed. EUROMECH Colloquium 524, Enschede, Netherlands, February 27-29, 2012.

## Research stays

- April - September 2012 with Michael Sherman and Scott Delp. Neuromuscular Biomechanics Lab (NMBL), Stanford University, Stanford, USA.

## Posters

- I. Stavness, **A. Scholz**, M. Sherman, A. Kecskeméthy, S. Delp. A general formulation of muscle wrapping for musculoskeletal simulations. Biomechanical Engineering Conference at Stanford (BMECS), May 22, 2012.

## Research talks

- Helmholtz-Institut für Biomedizinische Technik, RWTH Aachen University, March 17, 2015.

## Theses

- A procedure for continuously differentiable muscle wrapping over many bodies for musculoskeletal simulation. Master Thesis, University of Duisburg-Essen, Duisburg, Germany, in cooperation with Stanford University, Stanford, USA, 2012.
- Trajectory planning optimization of cooperating mechanisms under dynamic constraints. Bachelor Thesis, University of Duisburg-Essen, Duisburg, Germany, 2011.

# Bibliography

[1] P. Stäckel. Bemerkungen über die Geschichte der geodätischen Linien. *Berichte sächs. Akad. Wiss.*, 45:444–467, 1893.

[2] P. Sager. Übersicht über die Entwicklung der geodätischen Linien seit Gauß. *Landes-Universität Rostock*, 1903.

[3] K. Reich. Die Geschichte der Differentialgeometrie von Gauß bis Riemann (1828-1868). *Technische Universität München*, 1973.

[4] H. v. Helmholtz. Über den Stoffverbrauch bei der Muskelaktion. *J. Müller*, 1845.

[5] R. Heidenhain. Mechanische Leistung, Wärmeentwicklung und Stoffumsatz bei der Muskelthätigkeit. Ein Beitrag zur Theorie der Muskelkräfte. *Breitkopf und Härtel*, 1864.

[6] A. Fick. Untersuchungen über Muskel-Arbeit. *H. Georg's Verlag*, 1867.

[7] A. V. Hill. The absolute mechanical efficiency of the contraction of an isolated muscle. *J. Physiol.*, 46(6):435–469, 1913.

[8] A. V. Hill. The maximum work and the mechanical efficiency of human muscles, and their most economical speed. *J. Physiol.*, 56(1-2):19–41, 1922.

[9] A. V. Hill. The heat of shortening and the dynamic constants of muscle. *Proc. Roy. Soc. Biol. Sc.*, 126:612–745, 1938.

[10] A. V. Hill. The mechanics of active muscle. *Proc. Roy. Soc. Biol. Sc.*, 141:104–117, 1953.

[11] A. Fick. Die medicinische Physik. *Vieweg*, 1885.

[12] H. S. Gasser and A. V. Hill. The dynamics of muscular contraction. *Proc. Roy. Soc. Biol. Sc.*, 96(678):398–437, 1924.

[13] M. Millard, T. Uchida, A. Seth, and S. L. Delp. Flexing computational muscle: modeling and simulation of musculotendon dynamics. *J. Biomech. Eng.*, 135:021004–1 – 021004–11, 2013.

[14] A. F. Huxley and R. Niedergerke. Structural changes in muscle during contraction: inteference microscopy of living muscle fibres. *Nature*, 173:971–973, 1954.

[15] H. E. Huxley and J. Hanson. Changes in the cross-striations of muscle during contraction and stretch and their structural interpretation. *Nature*, 173:973–976, 1954.

[16] A. F. Huxley. Muscle structure and theories of contraction. *Prog. Biophys. Biophys. Chem.*, 7:255–318, 1957.

[17] A. M. Gordon, A. F. Huxley, and F. J. Julian. The variation in isometric tension with sarcomere length in vertebrae muscle fibers. *J. Physiol.*, 184:170–192, 1966.

[18] A. F. Huxley and R. M. Simmons. Proposed mechanism of force generation in striated muscle. *J. Physiol.*, 233:533–583, 1971.

[19] J. C. Haselgrove and H. E. Huxley. X-ray evidence for radial cross-bridge movement and for the sliding filament model in actively contracting skeletal muscle. *J. Mol. Biol.*, 77:549–568, 1973.

[20] E. Eisenberg, T. L. Hill, and Y. Chen. Cross-bridge model of muscle contraction. quantitative analysis. *J. Biophys.*, 29(2):195–227, 1980.

[21] G. Zahalak and S.-P. Ma. Muscle activation and contraction: constitutive relations based directly on cross-bridge kinetics. *J. Biomech. Eng.*, 112(1):52–62, 1990.

[22] A. S. Arnold, S. S. Blemker, and S. L. Delp. Evaluation of a deformable musculoskeletal model for estimating muscle-tendon lengths during crouch gait. *Ann. Biomed. Eng.*, 29:263–274, 2001.

[23] G. H. Kerr and P. Selber. Musculoskeletal aspects of cerebral palsy. *J. Bone Joint Surg. - Brit. Vol.*, 85(2):157–166, 2003.

[24] A. S. Arnold, F. C. Anderson, M. G. Pandy, and S. L. Delp. Muscular contributions to hip and knee extension during the single limb stance phase of normal gait: a framework for investigating the causes of crouch gait. *J. Biomech.*, 38:2181–2189, 2005.

[25] J. L. Hicks, M. H. Schwartz, A. S. Arnold, and S. L. Delp. Crouched postures reduce the capacity of muscles to extend the hip and knee during the single-limb stance phase of gait. *J. Biomech.*, 41:960–967, 2008.

[26] K. M. Steele, A. Seth, J. L. Hicks, M. S. Schwartz, and S. L. Delp. Muscle contributions to support and progression during single-limb stance in crouch gait. *J. Biomech.*, 43:2099–2105, 2010.

[27] R. R. Neptune, S. A. Kautz, and F. E. Zajac. Contributions of the individual ankle plantar flexors to support, forward progression and swing initiation during walking. *J. Biomech.*, 34:1387–1398, 2001.

[28] F. E. Zajac, R. R. Neptune, and S. A. Kautz. Biomechanics and muscle coordination of human walking part 1: introduction to concepts, power transfer, dynamics and simulations. *Gait Posture*, 16:215–232, 2002.

[29] F. E. Zajac, R. R. Neptune, and S. A. Kautz. Biomechanics and muscle coordination of human walking part 2: lessons from dynamical simulations and clinical implications. *Gait Posture*, 17:1–17, 2003.

[30] M. Q. Liu, F. C. Anderson, M. G. Pandy, and S. L. Delp. Muscles that support the body also modulate forward progression during walking. *J. Biomech.*, 39:2623–2630, 2006.

[31] M. M. van der Krogt, S. L. Delp, and M. H. Schwartz. How robust is human gait to muscle weakness? *Gait Posture*, 36:113–119, 2012.

[32] R. R. Neptune and K. Sasaki. Ankle plantar flexor force production is an important determinant of the preferred walk-to-run transition speed. *J. Exp. Biol.*, 208:799–808, 2005.

[33] S. R. Hamner, A. Seth, and S. L. Delp. Muscle contributions to propulsion and support during running. *J. Biomech.*, 43:2709–2716, 2010.

[34] F. C. T. van der Helm. The shoulder mechanism: a dynamic approach. *Delft University of Technology*, 1991.

[35] J. Yu, D. C. Ackland, and M. G. Pandy. Shoulder muscle function depends on elbow joint position: an illustration of dynamic coupling in the upper limb. *J. Biomech.*, 44:1859–1868, 2011.

[36] K. Sasaki and R. R. Neptune. Individual muscle contributions to the axial knee joint contact force during normal walking. *J. Biomech.*, 43:2780–2784, 2010.

[37] Y.-C. Lin, J. P. Walter, S. A. Banks, M. G. Pandy, and B. J. Fregly. Simultaneous prediction of muscle and contact forces in the knee during gait. *J. Biomech.*, 43:945–952, 2010.

[38] C. R. Winby, D. G. Lloyd, T. F. Besier, and T. B. Kirk. Muscle and external load contribution to knee joint contact loads during normal gait. *J. Biomech.*, 42:2294–2300, 2009.

[39] F. Moissenet, L. Chèze, and R. Dumas. A 3D lower limb musculoskeletal model for simultaneous estimation of musculo-tendon, joint contact, ligament and bone forces during gait. *J. Biomech.*, 2013.

[40] Y. Giat, J. Mizrahl, W. S. Levine, and J. Chen. Simulation of distal tendon transfer of the biceps brachii and the brachialis muscles. *J. Biomech.*, 27(8):1005–1014, 1994.

[41] R. A. Brand, R. D. Crowninshield, C. E. Wittstock, D. R. Pedersen, C. R. Clark, and F. M. van Krieken. A model of lower extremity muscular anatomy. *J. Biomech. Eng.*, 104(4):304–310, 1982.

[42] D. C. Ackland and M. G. Pandy. Lines of action and stabilizing potential of the shoulder musculature. *J. Anat.*, 215:184–197, 2009.

[43] S. S. Blemker, D. S. Asakawa, G. E. Gold, and S. L. Delp. Image-based musculoskeletal modeling: applications, advances, and future opportunities. *J. Magn. Reson. Im.*, 25:441–451, 2007.

[44] J. D. Webb, S. S. Blemker, and S. L. Delp. 3D finite element models of shoulder muscles for computing lines of actions and moment arms. *Comput. Methods Biomech. Biomed. Engin.*, 17(8):1–9, 2014.

[45] K.-A. An, L. Berglund, W. P. Cooney, E. Y. S. Chao, and N. Kovacevic. Direct in vivo tendon force measurement system. *J. Biomech.*, 23(12):1269–1271, 1990.

[46] F. Schuind, M. Garcia-Elias, W. P. Cooney III, and K.-N. An. Flexor tendon forces: in vivo measurements. *J. Hand Surg.*, 17(2):291–298, 1992.

[47] S. S. Blemker and S. L. Delp. Rectus femoris and vastus intermedius fiber excursions predicted by three-dimensional muscle models. *J. Biomech.*, 39:1383–1391, 2006.

[48] O. Röhrle, J. B. Davidson, and A. J. Pullan. Bridging scales: a three-dimensional electromechanical finite element model of skeletal muscle. *SIAM J. Sci. Comput.*, 30(6):2882–2904, 2008.

[49] P. Favre, C. Gerber, and J. G. Snedeker. Automated muscle wrapping using finite element detection. *J. Biomech.*, 43:1931–1940, 2010.

[50] L. A. Spyrou and N. Aravas. Muscle-driven finite element simulation of human foot movements. *Comput. Methods Biomech. Biomed. Engin.*, 5(9):925–934, 2012.

[51] R. R. Lemos, J. Rocke, G. V. G. Baranoski, Y. Kawakami, and T. Kurihara. Modeling and simulating the human skeletal muscle based on anatomy and physiology. *Comput. Animation Virtual Worlds*, 16:319–330, 2005.

[52] O. Röhrle and A. J. Pullan. Three-dimensional finite element modelling of muscle forces during mastication. *J. Biomech.*, 40:3363–3372, 2007.

[53] D. K. Pai. Muscle mass in musculoskeletal models. *J. Biomech.*, 43(11):2093–2098, 2010.

[54] C. J. Gatti and R. E. Hughes. Optimization of muscle wrapping objects using simulated annealing. *Ann. Biomed. Eng.*, 37(7):1342–1347, 2009.

[55] B. A. Garner and M. G. Pandy. The obstacle-set method for representing muscle paths in musculoskeletal simulations. *Comput. Methods Biomech. Biomed. Engin.*, 3:1–30, 1999.

[56] A. Audenaert and E. Audenaert. Global optimization method for combined spherical-cylindrical wrapping in musculoskeletal upper limb modeling. *Comput. Meth. Prog. Bio.*, 92:8–19, 2008.

[57] F. Gao, M. Damsgaard, J. Rasmussen, and S. T. Christensen. Computational method for muscle-path representation in musculoskeletal models. *Biol. Cybern.*, 87:199–210, 2002.

[58] E. M. Arnold, S. R. Ward, R. L. Lieber, and S. L. Delp. A model of the lower limb for analysis of human movement. *Ann. Biomed. Eng.*, 38(2):269–279, 2010.

[59] A. B. Carman and P. D. Milburn. Dynamic coordinate data for describing muscle-tendon paths: a mathematical approach. *J. Biomech.*, 38:943–951, 2005.

[60] I. I. Esat and N. Ozada. Articular human joint modeling. *Robotica*, 28:321–339, 2010.

[61] I. W. Charlton and G. R. Johnson. Application of spherical and cylindrical wrapping algorithms in a musculoskeletal model of the upper limb. *J. Biomech.*, 34:1209–1216, 2001.

[62] A. N. Vasavada, R. A. Lasher, T. E. Meyer, and D. C. Lin. Defining and evaluating wrapping surfaces for MRI-derived spinal muscle paths. *J. Biomech.*, 41:1450–1457, 2008.

[63] E. Desailly, P. Sardain, N. Khouri, D. Yepremian, and P. Lacouture. The convex wrapping algorithm: a method for identifying muscle paths using the underlying bone mesh. *J. Biomech.*, 43:2601–2607, 2010.

[64] G. E. Marai, D. H. Laidlaw, C. Demiralp, S. Andrews, C. M. Grimm, and J. J. Crisco. Estimating joint contact areas and ligaments lengths from bone kinematics and surfaces. *IEEE Trans. Biomed. Eng.*, 51(5):790–799, 2004.

[65] S. P. Marsden, D. C. Swailes, and G. R. Johnson. Algorithms for exact multi-object muscle wrapping and application to the deltoid muscle wrapping around the humerus. *Proc. Inst. Mech. Eng. H J. Eng. Med.*, 222(7):1081–1095, 2008.

[66] I. Stavness, M. Sherman, and S. L. Delp. A general approach to muscle wrapping over multiple surfaces. *Proc. Amer. Soc. Biomech., Florida, USA*, 2012.

[67] A. Scholz, I. Stavness, M. Sherman, S. L. Delp, and A. Kecskeméthy. Improved muscle wrapping algorithms using explicit path-error Jacobians. *Comput. Kinematics, Barcelona, Spain*, 2013.

[68] A. S. Bahler, J. T. Fales, and K. L. Zierler. The dynamic properties of mammalian skeletal muscle. *J. Gen. Physiol.*, 51:369–384, 1968.

[69] D. J. Struik. Lectures on classical differential geometry. *Dover Publications*, 1988.

[70] M. P. do Carmo. Differential geometry of curves and surfaces. *Prentice Hall*, 1976.

[71] A. Pressley. Elementary differential geoemetry. *Springer*, 2010.

[72] K. Strubecker. Differentialgeometrie Band 2: Theorie der Flächenmetrik. *Walter de Gruyter & Co*, 1969.

[73] M. P. do Carmo. Riemannian geometry. *Birkhäuser*, 1993.

[74] M. Spivak. A comprehensive introduction to differential geometry. *Publish or Perish*, 1999.

[75] R. Goldman. Curvature formulas for implicit curves and surfaces. *Comput. aided geom. d.*, 22:632–658, 2005.

[76] A. Kecskeméthy and M. Hiller. An object-oriented approach for an effective formulation of multibody dynamics. *Comp. Method. Appl. M.*, 115(3-4):287–314, 1994.

[77] A. C. Hindmarsh. OEPACK, a systematized collection of ODE solvers. *IMACS Trans. Scientific Comput.*, 1:55–64, 1983.

[78] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, and D. Sorensen. LAPACK: a portable linear algebra library for high-performance computers. *ACM/IEEE Conference on Supercomputing*, pages 2–11, 1990.

[79] S. L. Delp, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen. Opensim: open-source software to create and analyze dynamic simulations of movement. *Biomed. Eng.*, 54(11):1940–1950, 2007.

[80] K. Strubecker. Differentialgeometrie Band 1: Kurventheorie der Ebene und des Raumes. *Walter de Gruyter & Co*, 1964.

[81] K. Strubecker. Differentialgeometrie Band 3: Theorie der Flächenkrümmung. *Walter de Gruyter & Co*, 1969.

[82] F. E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Rev. in Biomed. Eng.*, 17(4):359–411, 1989.

[83] S. K. Semwal and J. J. Hallauer. Biomechanical modeling: implementing line-of-action algorithm for human muscles and bones using generalized cylinders. *Comput. & Graph.*, 18(1):105–112, 1994.

[84] S. L. Delp and P. Loan. A graphics-based software system to develop and analyze models of musculoskeletal structures. *Comput. Biol. Med.*, 25(1):21–34, 1995.

[85] N. Wuelker, C. J. Wirth, W. Plitz, and B. Roetman. A dynamic shoulder model: reliability testing and muscle force study. *J. Biomech.*, 28(5):489–499, 1995.

[86] D. K. Pai. Strands: interactive simulation of thin solids using cosserat models. *Eurographics*, 21(3), 2002.

[87] F. C. Anderson and M. G. Pandy. Individual muscle contributions to support in normal walking. *Gait Posture*, 17:159–169, 2003.

[88] A. N. Ahn, R. J. Monti, and R. J. Biewener. In vivo and in vitro heterogeneity of segment length changes in the semimembranosus muscle of the toad. *J. Phys.*, 549:877–888, 2003.

[89] J. Kruidhof and M. G. Pandy. Effect of muscle wrapping on model estimates of neck muscle strength. *Comput. Methods Biomech. Biomed. Engin.*, 9(6):343–352, 2006.

[90] S. P. Marsden and D. C. Swailes. A novel approach to the prediction of musculotendon paths. *Proc. Inst. Mech. Eng. H J. Eng. Med.*, 222(1):51–61, 2008.

[91] E. K. Chadwick, D. Blana, A. J. van den Bogert, and R. F. Kirsch. A real-time, 3D musculoskeletal model for dynamic simulation of arm movements. *IEEE Trans. Biomed. Eng.*, 56(4):941–948, 2009.

[92] E. Desailly, P. Sardain, N. Khouri, D. Yepremian, and P. Lacouture. The convex wrapping algorithm. *Comput. Methods Biomech. Biomed. Engin.*, 14:237–239, 2011.

[93] D. K. Pai, S. Sueda, and D. I. W. Levin. Dynamics of biomechanisms: musculotendon mass, constraints, and architecture. *IUTAM Symp. Human Body Dyn., Waterloo, Canada*, 2:158–167, 2011.

[94] A. J. von den Bogert, D. Blana, and D. Heinrich. Implicit methods for efficient musculoskeletal simulation and optimal control. *IUTAM Symp. Human Body Dyn., Waterloo, Canada*, 2:297–316, 2011.

[95] H. Thielhelm, A. Vais, D. Brandes, and F.-E. Wolter. Connecting geodesics on smooth surfaces. *Visual Comput.*, 28:529–539, 2012.

[96] B. L. Suderman and A. N. Vasavada. Moving muscle points provide accurate curved muscle paths in a model of the cervical spine. *J. Biomech.*, 45:400–404, 2012.

[97] A. Scholz, M. Sherman, I. Stavness, S. L. Delp, and A. Kecskeméthy. A fast multi-obstacle muscle wrapping method using natural geodesic variations. *Mulibody Syst. Dyn.*, 36(2):195–219, 2016.